ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra řídicí techniky

Diplomová práce

Dálkové řízení modelu

Praha, 2005 Perman Pavel

Prohlášení

Prohlašuji, že jsem svou diplomovou (bakalářskou) 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 ……. …….
podpis

Poděkování

Touto cestou bych chtěl rád poděkovat především vedoucímu své diplomové práce Doc. Ing. Jiřímu Bayerovi, CSc za cenné rady a připomínky při realizaci této práce. Dále bych rád poděkoval Ing. Pavlu Růžičkovi za konzultace, cenné připomínky a za čas, který mi věnoval při vzniku této práce. V neposlední řadě bych velice rád poděkoval svým rodičům a přátelům, kteří mě v celém průběhu studia podporovali.

Abstrakt

Tato diplomová práce popisuje realizaci dálkového řízení reálného modelu vzducholodi pomocí technologie Bluetooth. Bezdrátové, obousměrné komunikační spojení bylo zprostředkováno pomocí vestavných modulů Bluetooth. K řízení modulů bylo využito na jedné straně osobního počítače a na straně druhé bylo užito vestavného (embedded) mikrokontroléru.

Komunikačního spojení je využito pro přenos dat z reálného modelu vybaveného senzory pro orientaci v neznámem prostředí. Získaná data jsou v hlavním počítači zpracována a zobrazena jednoduchým algoritmem. Takto naměřená data jsou využita k řízení modelu tak, aby se model bezpečně pohyboval v neznámém prostoru a zamezil tak případné srážce s překážkou.

Abstract

This thesis describes realization of remote control of real model airship base on technology Bluetooth. For wireless, full-duplex communication was used embedded modules of Bluetooth. One operating module was utilized on one's parties personal computer and on the other side was used embedded microcontroller.

Communication is utilized for data transmission from real model equips sensors for orientation in unknown space. Acquired data are processed in master computer. This data are representit simple algorithm. This measuring data are utilized to drive model in order to control this model in safe spaces and exclude probabilitty collision with barriers.

Obsah

1. Kapitola 1
Úvod 1
2. Kapitola 2
Bluetooth technologie 2
2.1 Úvod do technologie Bluetooth 2
2.1.1 Historie Bluetooth 2
2.1.2 Stručný popis technologie Bluetooth 3
2.2 Jednotlivé části specifikace 4
2.2.1 Radiová HW vrstva (Bluetooth Radio) 4
2.2.2 Baseband 5
2.2.3 Protokol linkového manageru (LMP) 20
2.2.4 Vrstva HCI (Host Controller Interfaces) 21
2.2.5 Vyšší vrstvy zásobníku 26
3. Kapitola 28
Popis modelu a návrh hardwarových částí 28
3.1 Model vzducholodi 28
3.1.1 Jednotky pohonů modelu 29
3.1.2 Zařízení pro orientaci v prostoru 30
3.1.3 Řídicí jednotka modelu 30
3.2 Popis modulu SieMo – S50037 32
3.2.1 Vlastnosti modulu SieMo 32
3.2.2 Rozhraní modulu SieMo S50037 33
3.2.3 Primární konfigurační databáze (Persistent Store Key) 34
3.3 Návrh hardwarové části 35
3.3.1 Připojení modulu SieMo k osobnímu počítači 36
3.3.2 Připojení modulu SieMo k mikrokontroléru PIC 37
3.3.3 Vývojové zařízení pro modul Bluetooth 39
3.3.4 Mikrokontrolér PIC16F873 40

4. Kapitola 44
Návrh a implementace komunikačního softwaru 44
4.1 Popis komunikačního protokolu modelu 44
4.2 Program pro mikrokontrolér PIC 16F873 45
4.2.1 Struktura programu 45
4.2.2 Popis knihovny BT_func.asm 47
4.3 Hlavní řídicí program 51
4.3.1 Přístup k sériovému portu 51
4.3.2 Struktura řídicího programu modelu 52
4.4 Program pro komunikaci mezi dvěma počítači 56
4.4.1 Popis struktury programu 56
5. Kapitola 58
Návrh řízení modelu vzducholodi 58
5.1 Popis systému 58
5.1.1 Měření přechodových charakteristik a získání přenosu 59
5.1.2 Návrh regulátoru 64
5.1.3 Využití naměřených dat k orientaci v prostředí 66
6. Kapitola 68
Závěr 68
Dodatek A 70
Dodatek B 72
Dodatek C 75
Reference 77

Seznam obrázků

Obr. 1 protokolový zásobník. 4
Obr. 2 modulace FSK. 5
Obr. 3 základní koncepce Bluetooth. 6
Obr. 4 podporované uspořádání sítí, a), b) piconet (single master), c) scatternet (multimaster). 6
Obr. 5 časové schéma TDD (Time – Division - duplex). 7
Obr. 6 víceslotové časování paketů. 8
Obr. 7 unikátní BD adresa zařízení Bluetooth. 10
Obr. 8 základní typ baseband paketu. 12
Obr. 9 formát paketu Access code. 12
Obr. 10 formát hlavičky standardního paketu baseband vrstvy. 13
Obr. 11 sekvence kroků pro navázání spojení. 17
Obr. 12 diagram přechodů mezi jednotlivými stavy (Link Controller). 20
Obr. 13 komunikace na úrovní linkového manageru. 20
Obr. 14 navázání spojení pomocí LMP. 21
Obr. 15 propojení jednotlivých vrstev HCI. 22
Obr. 16 formát paketu HCI příkazu. 24
Obr. 17 vytvoření *OpCodu* HCI příkazu. 24
Obr. 18 formát paktu HCI událostí. 25
Obr. 19 formát datového paketu. 25
Obr. 20 model vzducholodi. 28
Obr. 21 řízení polohového servopohonu pomocí PWM. 29
Obr. 22 možnosti natočení hlavního pohonu. 29
Obr. 23 blokové schéma řídicí části modelu. 31
Obr. 24 modul SieMo S50037. 34
Obr. 25 program PStool pro konfiguraci primárních parametrů. 35
Obr. 26 schéma zapojení Bluetooth modulu k osobnímu počítači. 36
Obr. 27 grafická reprezentace rámce pro sériovou komunikaci. 37
Obr. 28 připojení modulu Bluetooth k řídicímu mikrokontroléru. 38
Obr. 29 popis jednotlivých pinů mikrokontroléru PIC 16F873. 41

Obr. 30 přímé a nepřímé adresování v PIC 16F873. 43
Obr. 31 organizace programové paměti a zásobníku. 43
Obr. 32 standardní formát příkazu pro řízení modelu. 44
Obr. 33 standardní typ odpovědi na příkaz pro řízení modelu. 44
Obr. 34 struktura programu pro PIC 16F873. 46
Obr. 35 definice a vložení makra do programu. 47
Obr. 36 vývojový diagram makra SET_BT_SLAVE. 49
Obr. 37 vývojový diagram makra *CONNECTION?*. 50
Obr. 38 způsob implementace tabulky HCI příkazů. 51
Obr. 39 hlavní formulář aplikace. 54
Obr. 40 struktura hlavního řídicího programu. 55
Obr. 41 struktura programu pro komunikaci mezi počítači. 57
Obr. 42 přechodová charakteristika výšky modelu. 61
Obr. 43 normovaná přechodová charakteristika výšky modelu. 61
Obr. 44 přechodová charakteristika natočení modelu. 62
Obr. 45 normovaná přechodová charakteristika natočení modelu. 62
Obr. 46 přechodová charakteristika polohy v horizontálním směru. 63
Obr. 47 přechodová charakteristika modelu v horizontálním směru. 63
Obr. 48 porovnání normovaných přechodových charakteristik. 64
Obr. 49 výsledek regulace výšky. 65
Obr. 50 regulace natočení modelu (odchylka od severu). 65
Obr. 51 výsledný obraz okolního prostředí. 67

Seznam tabulek

Tab. 1 třídy zařízení Bluetooth. 3
Tab. 2 rozsahy frekvencí podle území. 5
Tab. 3 typy HCI paketů. 23
Tab. 4 skupiny HCI příkazů. 23
Tab. 5 napěťové úrovně jednotlivých zařízení. 37
Tab. 6 nastavení komunikačních parametrů rozhraní UART. 38
Tab. 7 závislost umístění inflexního bodu na řádu systému. ..... **Chyba! Záložka není definována.**

1. Kapitola

Úvod

Tato práce se zabývá dálkovým řízením mobilního zařízení. Mobilní zařízení představuje reálný model vzducholodi, který bude ovládán jak manuálně tak automaticky. Oba tyto režimy řízení bude zajišťovat osobní počítač. Stěžejní částí této práce je tedy zprostředkovat vzájemnou bezdrátovou komunikaci reálného modelu s osobním počítačem. Bezdrátová komunikace je podle zadání řešena pomocí technologie Bluetooth.

Seznámení s vlastní funkcí, vlastnostmi modulu *SieMo* a protokolem Bluetooth bude ověřeno pomocí demonstračního programu, zajišťující spojení mezi dvěma počítači PC. Demonstrační program musí umožnit navázání spojení mezi oběma moduly, dále pak vytvořit komunikační kanál, přes který bude možné přenést předem stanovená data a spojení ukončit. Popisem protokolu Bluetooth zabývá kapitola 2.

Následným úkolem je realizace spojení mezi osobním počítačem a mikrokontrolérem, který umožní komunikaci přes Bluetooth s řídicí stanicí a ovládaní všech periférií v reálném modelu. Model vzducholodi bude doplněn o čidla pro orientaci v prostoru, jejichž naměřená data budou následně zpracována řídicím počítačem. Jako čidla prostoru byly zvoleny ultrazvukové dálkoměry, elektronický kompas, akcelerometry a čidla náklonu pro kalibraci el. kompasu. Dále model obsahuje pohony pro řízení letu a power management pro kontrolu proudového odběru a napětí akumulátoru. Popis celkového modelu, doplněného o senzory a komunikační modul je uveden v kapitole 3. Implementací softwaru do mikrokontroléru a osobního počítače se pak zabývá kapitola 4.

Získaná data ze senzorů jsou využita pro řízení, výšky a natočení modelu v horizontální rovině. Na základě dat získaných z ultrazvukových dálkoměrů pro snímání okolního prostoru, řídicí počítač zrekonstruuje podle jednoduchého algoritmu mapu prostoru. Po lokalizaci volného prostoru, je vygenerováno povel k pohybu modelu (akční zásah). Tento akční zásah je odeslán přes bezdrátové rozhraní. Identifikací modelu, návrhem řízení a příslušných regulátorů se věnuje kapitola 5.

Hlavní úkol tedy spočívá v dálkovém řízení modelu tak, aby byl schopen se bezpečně pohybovat v neznámém prostředí a pomocí výše zmíněných čidel zamezil případné srážce.

2. Kapitola

Bluetooth technologie

Tato kapitola se bude věnovat popisu moderní bezdrátové technologie Bluetooth a jeho protokolu. Popis protokolu odpovídá verzi 1.1 (únor 2002), protože tato verze byla implementována ve modulu Bluetooth *SieMo*. Od listopadu 2003 je k dispozici verze 1.2 podporující vyšší rychlosti a v současné době je připravena i verze 2.0.

2.1 Úvod do technologie Bluetooth

Hlavní myšlenkou Bluetooth je vytvořit univerzální radiové rozhranní, pro vytvoření bezdrátového spojení na krátkou vzdálenost. Jde tedy o standard, který poskytne bezdrátového spojení mezi počítačem a jeho periferiemi. Byl navržen za účelem odstranění metalického vedení mezi periferiemi a osobním počítačem, používající rozšířené komunikační standardy. V návrhu byly upřednostněny klíčové vlastnosti jako nízká složitost, napájení a především nízká cena (předpoklad ceny se pohyboval kolem 5 USD za plnohodnotné zařízení).

Bluetooth byl navržen jako komunikační standard, který pro přenos dat využívá frekvenční pásmo ISM (Industrial, Scientific, Medical), jenž lze volně využívat za předpokladu dodržení závazných podmínek pro vyzářený výkon (tzv. nelicencované pásmo).Volné použití pásma znamená, že není třeba žádat o přidělení frekvenčního pásma ani platit jakékoliv poplatky.

2.1.1 Historie Bluetooth

Vývoj započala švédská společnost Ericsson. Připojení dalších organizací ke společnosti Ericsson dalo vzniknout konsorciu firem SIG (Special Interest Group), do kterých v počátku spadaly především významné společnosti jako 3Com, IBM, Intel, Microsoft, Motorola, Nokia a Toshiba. Společným úsilím pak v roce 1999 konsorcium představilo první verzi standardu Bluetooth 1.0. K roku 2003 čítá společenství zhruba 2000 firem, které se zajímají o vývoj a aplikace tohoto standardu. Bluetooth je první globálně akceptovaná

technologie kategorie PAN (*Personal Area Network*) a je zahrnuta do řady specifikací PAN IEEE 802.15 pod označením IEEE 802.15.1.

2.1.2 Stručný popis technologie Bluetooth

Jak již bylo naznačeno v úvodu jde o radiovou komunikaci na frekvenci 2.4 *GHz* v pásmu ISM, které je kromě Francie (užší pásmo) celosvětové bezlicenční pásmo. Jde o velice spolehlivou a bezpečnou komunikaci, odolnou proti rušení. Pro přenos dat je využíváno frekvenčního skákání (Frequency hopping). Název výstižně popisuje systém komunikace, při kterém dochází k velice rychlé změně vysílací frekvence. Bluetooth disponuje 79 kanály a je schopen provést až 1600 změn frekvence za jednu sekundu. Zařízení Bluetooth využívá aktuálně nastavenou sekvenci frekvenčních skoků, která je nastavena při vytvoření spojení. Tím je zajištěna ještě větší bezpečnost komunikace a odolnost proti rušení. Zařízení Bluetooth jsou členěna podle vysílacího výkonu (dosahu) do skupiny třech tříd (Class). Jednotlivé třídy s odpovídajícími vysílacími výkony jsou uvedeny v tab. 1.

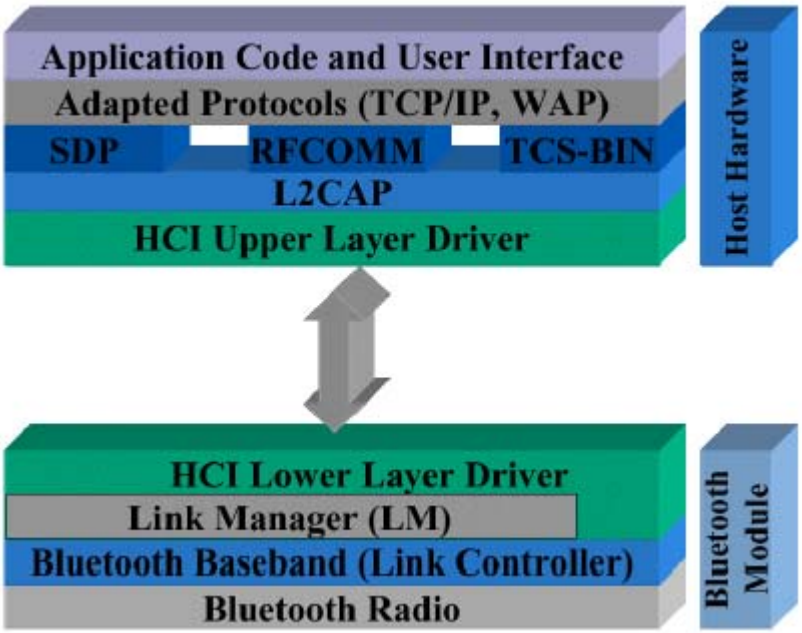
Třída zařízení (Class)	Max. výstupní výkon	Nom. výstupní výkon	Min. výstupní výkon	přibližný dosah
1	100 mW (20dBm)	není stanoveno	1 mW (0dBm)	100 m
2	2.5 mW (4dBm)	1 mW (0dBm)	0.25 mW (-6dBm)	20 m
3	1 mW (0 dBm)	není stanoveno	není stanoveno	1 m

Tab. 1 třídy zařízení Bluetooth.

Bluetooth standard je navržen jako komunikace typu master – slave, komunikace je tedy řízena prostřednictvím mastera jako hlavního prvku, který vždy iniciuje spojení. Tento standard podporuje i výměny rolí, tedy prohození funkcí mastera a slave. Master může zároveň komunikovat až se sedmi zařízeními typu slave, tato topologile je ve specifikaci [1] uvedena jako *piconet* (detailněji bude popsána v další části této kapitoly). Je-li některé zařízení typu slave vlastněno více mastery, což specifikace umožňuje, vytváří se tak topologická síť nazvaná podle [1] *scatternet* (detailněji v další kapitole).

2.2 Jednotlivé části specifikace

V této oddílu budou popsány všechny významné části specifikace technologie Bluetooth. Celý protokolový zásobník je uveden na obr.1. Nejvýznamnější vrstvy, které určují způsob komunikace jsou uloženy nejníže na zásobníku. Jde o tři vrstvy pod názvy *Bluetooth Radio*, *Baseband* a *LMP* (Link Manager Protokol). Přístup na tyto vrstvy, které jsou vždy implementovány v jádru komunikačních modulů, sjednocuje protokolová vrstva HCI (Host Controller Interface). Nad HCI jsou umístěny vrstvy pro zvýšení využitelnosti komunikačního rozhraní Bluetooth poskytující tak přístup a komunikaci s jinými standardy a protokoly.



Obr. 1 protokolový zásobník.

2.2.1 Radiová HW vrstva (Bluetooth Radio)

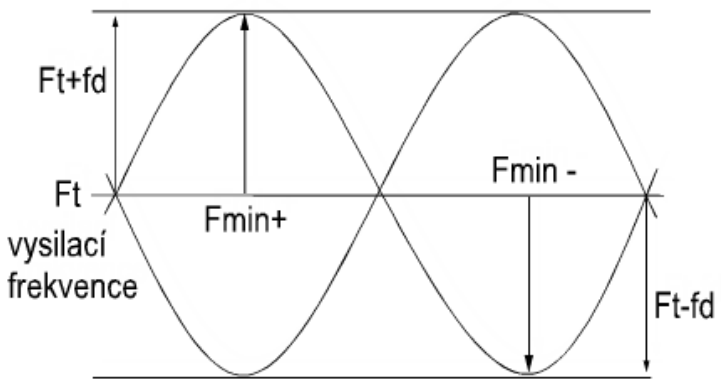
Tato vrstva definuje požadavky na vysokofrekvenční část) operující v nelicencovaném pásmu (ISM). Operační podmínky RF vysílače jsou definovány v [1]. Radiový vysílač pracuje ve frekvenčním rozsahu 2400 - 2483 *MHz* a obsahuje 79 RF kanálů. Frekvence jednotlivých kanálů jsou odstupňovány po 1 *MHz* a hodnoty jejich frekvencí jsou stanoveny podle tab 2. Na území Franci je frekvenční rozsah a počet využívaných kanálů omezen.

Oblast	Frek. Rozsah	Frek. RF kanálu	k
USA, Evropa	2.400 - 2.483 GHz	f = 2402 + k (MHz)	0 - 78
Francie	2.446 - 2.483 GHz	f = 2454 + k (MHz)	0 - 22

Tab. 2 rozsahy frekvencí podle území.

Modulace nosného signálu

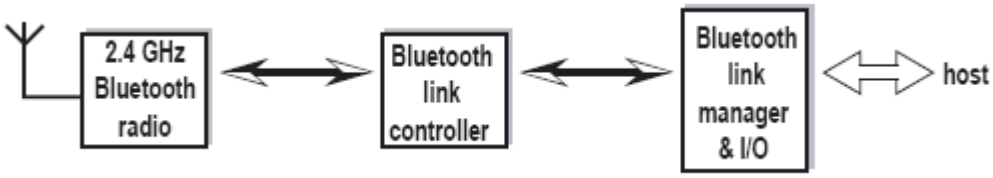
Výstupní vysílač nosného kmitočtu je ovlivňován pomocí GSFK (Gaussian Frequency Shift Keying) modulace. Gaussovská modulace s frekvenčním klíčování využívající modulátoru FSK, kterému je předřazena dolní propust, G filtr. Po průchodu pravoúhlého signálu tímto filtrem dojde ke omezení frekvenčního spektra směrem k vysokým kmitočtů a tím tedy ke snížení přenášeného frekvenčního rozsahu. FSK je modulace stavu nosného kmitočtu pomcí log. bitů. Logická úroveň je reprezentována kladnou (log. 1) resp. zápornou (log. 0) frekvenční odchylkou jak je vyznačeno na obr. 2. Jde o bezpečnou metodu přenosu. Nevýhodou je pouze vyšší energetická náročnost.



Obr. 2 modulace FSK.

2.2.2 Baseband

Popisuje fyzickou vrstvu technologie Bluetooth. Vysokofrekvenční vysílač pracující na frekvenci 2.4 *GHz* využívá frekvenčního skákání pro potkačení interference a dalšího možného rušení. Tato část protokolu zajišťuje také výběr frekvence pro frekvenční skákání a detekci chyb v přijatých paketech. Definuje také hlavní formát dat, fyzické kanály a adresování zařízení. Na této úrovni je také implementován *Link Controler*, zajišťující správu linek a navázání spojení. Zprávy linkové vrstvy jsou definovány v protokolu *LMP* (Link Manager Protokol). Základní koncepce jednotky Bluetooth je uvedena na obr. 3.

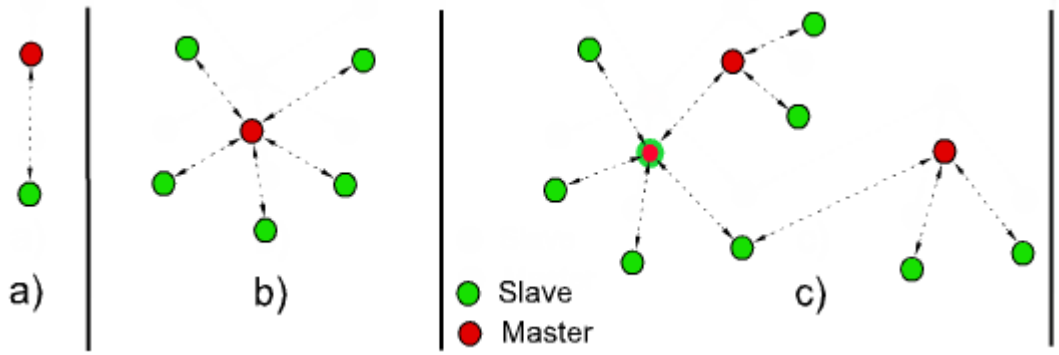


Obr. 3 základní koncepce Bluetooth.

Topologie a typy spojení

Jak již bylo výše uvedeno jde o spojení typu master – slave, přičemž master je schopen komunikovat až se sedmi podřízenými zařízeními. Protokol Bluetooth umožňuje navázat spojení bod – bod (point –to - point), při kterém jsou ve spojení pouze dvě zařízení. Nebo bod – více bodů (point – to – multipoint), při kterém všechna zúčastněná zařízení využívají stejný kanál. Toto topologické uspořádání je uvedeno na obr. 4a, 4b a nazývá se *piconet*. V tomto zřízení může být tedy maximálně sedm zařízení typu slave. Protokol umožňuje připojit i více podřízených zařízení k masterovi, využitím tzv. parkovacího stavu (Parked State). Jde o neaktivní připojení k masterovi, ale zachovává se synchronizace s *piconetem* a tedy i s řídicím masterem.

Každá síť (*piconet*) může být řízena pouze jedním masterem, ale slave může být vlastněn více mastery a tedy může spadat do oblasti jiného *piconetu*. Dále protokol umožňuje výměnu rolí master – slave (v čase kdy nekomunikují). Tyto vlastnosti pak umožňují topologické uspořádání nazvané *scatternet* (obr. 4c), které umožňují propojení více sítí *piconet*. Celková síť pak může obsahovat více masterů řídicí jednotlivé podsítě, které nejsou mezi sebou frekvenčně synchronizované. Každý *piconet* komunikuje na jiném kanálu s jinou sekvencí změn frekvence.

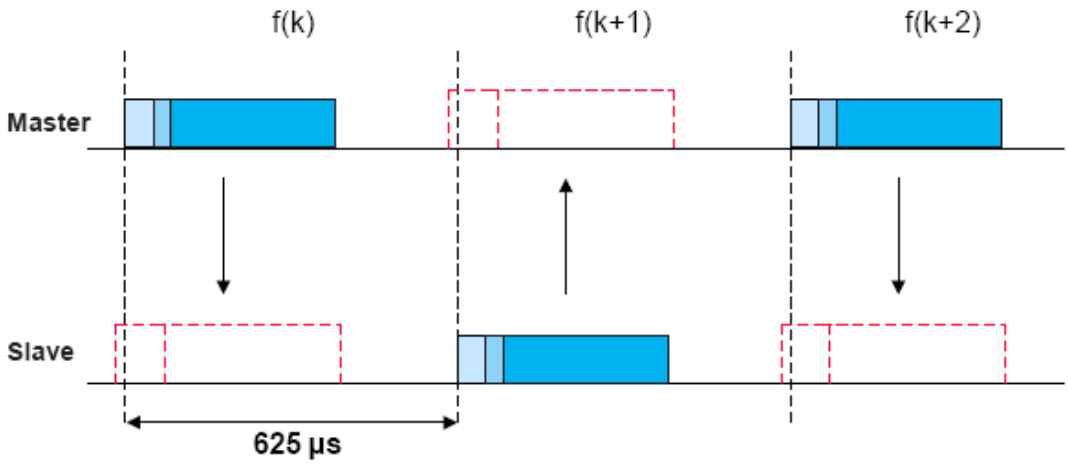


Obr. 4 podporované uspořádání sítí, a), b) piconet (single master), c) scatternet (multimaster).

Fyzické kanály

Fyzický kanál je definován jako pseudonáhodná sekvence frekvenčních skoků (sequence hopping), která využívá v časovém intervalu 625 μs vždy jeden ze 79 (23) dostupných frekvenčních kanálů. Toto frekvenční skákání je unikátní pro každý *piconet* a je nastaven masterem, který řídí daný *piconet*. Sekvence je odvozena od hodin mastera (Master Clock). Jelikož jde o frekvenční skoky, které se mění v časovém intervalu 625 μs, je vysílaní rozděleno do časových rámců (Time slots), které korespondují s frekvencí skoků.

Existují dva druhy časování kanálů. Jako základní pro plně full - duplexní provoz je využíváno TDD (Time – Division - Duplex) schématu. Časování TDD odpovídá situaci, v níž se řídicí a řízená jednotka postupně střídají ve vysílaní v každém časovém slotu. Schéma TDD je uvedeno na obr. 5.

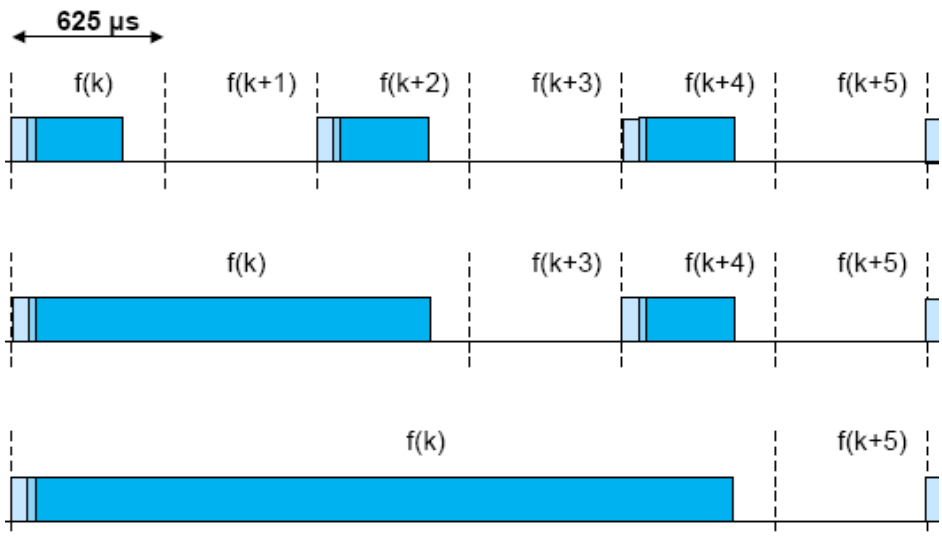


Obr. 5 časové schéma TDD (Time – Division - duplex).

Start přenosu by měl zahájit master vysíláním pouze v sudých časových slotech a slave by měl v tomto slotu přijímat. V časových slotech s lichým číslem je situace opačná. Během přenosu nesmí dojít ke změně frekvenčního kanálu.

Kromě přenosu v jednom slotu, specifikace podporuje přenosy víceslotové (multi-slots). Jde o přenosy, které zabírají pro vysílání dat více než jeden časový slot. Délka časového slotu může tedy být prodloužena na troj. až pětinásobek základního časového intervalu. V tomto prodlouženém časovém slotu musí být zajištěn pevný frekvenční kanál, musí být tedy pozdržena sekvence frekvenčních skoků. Násobky základního časového

intervalu musí být nutně liché, aby po skončení vysílaní mělo párové zařízení možnost vyslaný paket přijmout. Využití víceslotového časování je uvedeno na obr. 6.



Obr. 6 víceslotové časování paketů.

Fyzické linky (typy přenosů)

Standard bluetooth umožňuje využívat dva typy komunikačních přenosů, které se mezi sebou velice liší svými vlastnostmi. Definovány jsou dva typy:

- ACL (Asynchronous Connection - Less) – asynchronní spojení
- SCO (Synchronous Connection Oriented) – synchronní spojení

Obě tyto spojení lze využít k přenosu dat a je možné je v průběhu komunikace měnit.

SCO (Synchronous Connection Oriented)

Je symetrické spojení point – point mezi masterem a specifickým zařízením typu slave. Toto spojení podporuje přenos informací v přesně stanovených časových intervalech. Je tedy definováno maximální zpoždění přenášených informací. SCO spojení se tedy s výhodou využívá pro přenos hlasových či obrazových vzorků. Každé SCO spojení podporuje přenosovou rychlost 64 *kb/s* v obou směrech. Z důvodu časově kritického přenosu není u tohoto spojení povolena detekce a korekce chybného přenosu, tzn. že chybné pakety nejsou znovu přeposílány, ale jsou přímo nahrazeny novými daty. Při komunikaci přes SCO vždy

master vysílá v pravidelných časových intervalech a slave odpovídá v následujícím časovém slotu, tzv obvodové přepínaní (circuit-switched).

Master může podporovat až tři SCO linky k jednomu zařízení nebo k různým zařízením typu slave. Slave je také schopen komunikovat s masterem až po třech spojení SCO nebo může vytvořit SCO spojení pro dva rozdílné mastery. SCO komunikace je iniciována masterem, který odešle SCO nastaveí přes *LM* protokol (Link Manager Protokol). *LMP* protokol pro navázání SCO spojení obsahuje časové parametry jako jsou interval *Tsco* (délka pravidelného SCO intervalu) a *Dsco* (offset specifikující rezervovaný slot). Detailnější informace k SCO spojení jsou uvedeny [1].

ACL (Asynchronous Connection Less)

ACL linka využívá paketového přepínání mezi masterem a všemi aktivními zařízeními typu slave patřící do *piconetu*. ACL spojení mezi masterem a slavem může existovat pouze jedno. Oproti SCO spojení poskytuje ACL kontrolu integrity dat pomocí CRC. Při detekování chyby vyžádá přeposlání opravného paketu. To samozřejmě snižuje přenosovou rychlost. ACL linka podporuje buď symetrické spojení se shodnou přenosovou rychlostí v obou směrech. A to 433 *kb/s*. Nebo nesymetrické spojení s rychlostí 723 *kb/s* v jednom směru a 57.6 *kb/s* ve směru opačném. Využití dané linky závisí na velikosti zasílaných paketů. Počet vysílaných a přijímaných bajtů určuje typ ACL spojení. Je určeno podle obsazených časových slotů a také podle toho zda vyžadují detekci a korekci chyb.

Logické kanály

Bluetooth specifikace podporuje pět druhů logických kanálů:

- LC (Link Control) a LM (Link Manger) jako řídicí kanály
- UA/ UI (User Asynchronous / Izochronous data), US (User Synchronous data), definované pro uživatelský přístup.

LC kanál – řídicí kanál, je umístěn v části hlavičky přenášeného paketu. Přináší informace na úrovni řídicí linky jako ARQ (schéma pro opakování zpráv, při nepotvrzení příjemcem), řízení toku dat, charakteristiku přenášených dat. Je tedy součástí každé hlavičky všech paketů kromě ID paketu, který neobsahuje hlavičku.

LM kanál – řídicí kanál, který přenáší informace vyměněné na úrovni linkových manžerů zpráv mezi masterem a slavem. *LM* kanál je indikován L_CH (11 bitovým) kódem, umístěným v hlavičce datové části paketu.

UA/UI kanál – uživatelský kanál, který využívá především vrstva L2CAP (vrstva umístěná nad vrstvou HCI, pro transparentní přenos delších zpráv než podporuje vrstva baseband) pro fragmentaci paketů. Opět vlastní L_CH kód určující zda jde o startovní paket (L_CH = 10) či pakety následující (L_CH = 01).

US kanál – tento kanál je také užíván pro přenos uživatelských dat, ale synchronním způsobem, je tedy využíván SCO spojením.

Adresování zařízení

BD_ADDR: každé Bluetooth zařízení lze identifikovat pomocí jeho unikátní adresy. Ta je složena z 48 bitů, které lze rozdělit na několik částí. Tato adresa je odvozena od standardu IEEE 802. Rozdělení BD adresy je uvedeno na obr. 7. Nejvýznamnější část BD adresy je unikátní číslo přidělené společností, která Bluetooth zařízení vyrábí. Celý adresový prostor pak dosahuje možnosti adresování až 2^{32}.

LSB											MSB
company_assigned						company_id					
LAP						UAP		NAP			
0000	0001	0000	0000	0000	0000	0001	0010	0111	1011	0011	0101

Obr. 7 unikátní BD adresa zařízení Bluetooth.

- LAP (Low Address Part) obsahuje 24 bitů, jsou určeny společností.
- UAP (Upper Address Part) spolu s NAP tvoři identifikační číslo společnosti (8 bitů)
- NAP (Not Significant Part) obsahuje zbylých 16 bitů.

Bluetooth systém využívá tzv.přístupové kódy (Access codes), které jsou odvozené od BD adresy příslušného zařízení. Jsou definovány tři různé přístupové kódy:

- DAC (Device Access Code) přístupový kód zařízení, užívaný během *Page*, *Page scan* a *Page response* stavů (procedury pro spojení se vzdáleným zařízením).

- CAC (Channel Access Code) přístupový kód kanálu, který charakterizuje kanál užívaný v *piconetu* a druh všech výměn úvodních paketů.

- IAC (Inquiry Access Code) dotazovací přístupový kód, hlavní kód pro *Inquiry* (operace pro vyhledání vzdálených zařízení) operace.

Všechny kódy jsou odvozeny od adresové části LAP BD adresy. Přístupový kód je užívaný jako signalizace pro přijímač, že přicházejí nová data, dále pak pro časování synchronizace a kompenzaci hodinového offsetu (Clock Offset).

AM_ADDR: (Active Member Address) tato adresa existuje pouze dočasně. Jde o 3-bitovou adresu, která je přidělena každému zařízení Bluetooth patřícího v dané chvíli do existujícího *piconetu*. Tato adresa zaniká ve chvíli, kdy přestane být zařízení typu slave aktivní v daném *piconetu*. Master neobsahuje žádnou AM adresu. Obsahuje - li AM adresa samé nuly, jde o zprávu typu broadcast, tedy o zprávu pro všechna aktivní zařízení slave v *piconetu*.

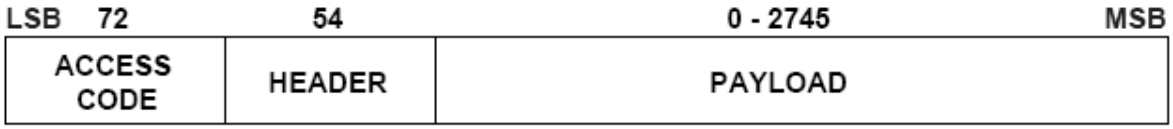
PM_ADDR: (Parked Member Address) adresu tvoří 8 bitů, které se adresují zařízení objevená masterem a zařazená do *piconetu*, ale nejsou aktivní. Jsou tedy v parkovacím stavu (Parked State). Tato adresa se danému zařízení přidělí v době, kdy přechází do stavu *Parked*. Spolu s přechodem zařízení slave do aktivního stavu, tato adresa zaniká a nahrazuje jí adresa AM_Addr.

AR_ADDR: (Access Request Address) tato adresa je používaná zařízením typ slave v módu *parked*, při synchronizaci se zařízením typu master. Nemusí být nutně unikátní.

Základní typy paketů

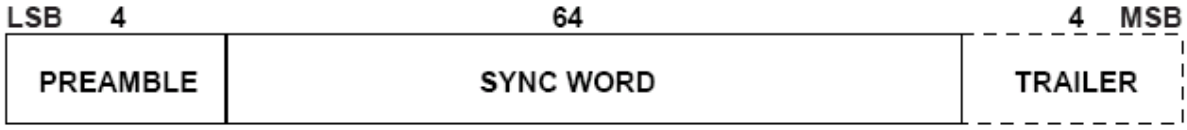
Všechny pakety, které definuje vrstva baseband využívají pořadí bitů v *Little endian* formátu. Toto uskupení bitů má na nejvýznamnějším místě (MSB) bit s nejméně významnou hodnotou (LSB), který je odesílán vždy jako první. Tato vrstva definuje 12 různých typů paketů. Pro vytvoření všech vyšších vrstev je využíváno těchto paketů. Popis těchto paketů bude uveden níže.

Data v *piconetu* jsou přenášena pomocí paketů. Formát hlavního paketu, který je využíván vrstvou baseband je na obr.8.



Obr. 8 základní typ baseband paketu.

Každý paket obsahuje tři části: *Access code* (72 bitů), *header* (54 bitů) a *payload* (proměnné). Typy a význam jednotlivých přístupových kódů byl již uveden výše. Na obr.9 je uveden jejich rámec. Část *payload* je určena pro přenášená data.

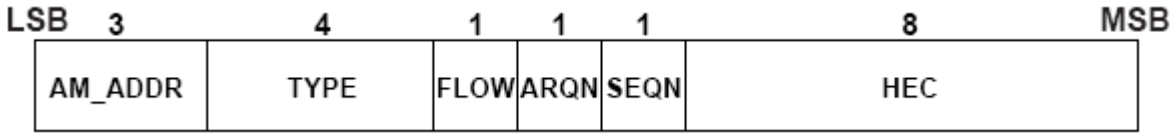


Obr. 9 formát paketu Access code.

Rámec paketu *Access Code* obsahuje úvodní část (preamble), synchronizační slovo (sync word) a přídavnou část (trailer). *Preamble* je složeno z fixně stanovených jedniček a nul. Tvoří ho buď vzorek 1010 nebo 0101 podle toho jak začíná synchronizační slovo. Synchronizační slovo je 64 bitové slovo odvozené od LAP adresy zařízení Bluetooth. Konstrukce garantuje největší možnou Hammingovu vzdálenost mezi synchronizačními slovy založenými na různých LAP adresách. *Trailer* je přídavná část stejného významu jako *Preamble*.

Hlavička paketu je složena *LC* (Link Control) informací a obsahuje 6 polí (obr. 10):

- AM_Addr: (3 bity) adresa aktivního zařízeni (slave) v *piconetu*.
- TYPE: (4 bity) rozlišuje sedm různých paketů, určuje který byl použit. Interpretace tohoto kódu závisí na fyzické lince (ACL, SCO).
- FLOW: (1 bit) tento bit je užíván k řízení toku pro pakety přenášené přes ACL. V případě, že je FLOW =1 je přijímací buffer prázdný a naopak.
- ARQN (1 bit) tento bit je určen pro potvrzování zpráv. Je-li 1 pak byla všechna data umístěna v datové části v pořádku přenesena.
- SEQN (1 bit) pro sekvenční číslování. Pro každý nově přenesený datový paket s CRC je tento bit invertován.
- HEC (8 bitů) kontrola chyb v hlavičce, zajištění integrity.



Obr. 10 formát hlavičky standardního paketu baseband vrstvy.

Vrstva baseband definuje 12 různých paketů pro obě fyzické linky (ACL, SCO). Další 4 řídicí pakety jsou pro obě linky společné. Detailní popis všech paketů je uveden ve specifikaci [1], zde budou uvedeny jen důležité pakety, jejichž význam je nutné znát i při práci s vyššími vrstvami (např. HCI).

Pro identifikaci typů paketů slouží 4 – bitový kód. Všechny pakety jsou rozděleny na čtyři skupiny. První skupina obsahuje čtyři řídicí pakety společné pro obě fyzické linky (NULL, POLL, FHS, DM1). Druhá skupina obsahuje šest paketů, které při přenosu zabírají jeden časový sloty. Další skupina paketů při přenosu zabírá tři časové sloty a poslední skupina pět časových slotů. Pro ACL spojení jsou definované pakety ve všech třech skupinách (DH1, AUX1, DM3, DH3, DM5, DH5). Pro fyzickou linku SCO jsou pakety definovány jen v první skupině. Všechny datové pakety tohoto spojení tedy zabírají pouze jeden časový slot. Souvisí to s vlastnostmi toho spojení (časově deterministické spojení).

Vzhledem k náplni této práce nemají typy paketů SCO žádný význam (nebyly využity), budou zde popsány jen typy paketů fyzické linky ACL, která byla využita pro komunikaci a pakety určené k řízení. Souhrnný přehled všech typů paketů s obsazením příslušných časových slotů je uveden v dodatku B.1.

Řídicí pakety

NULL – je užíván pro návrat informací k vysílací straně s ohledem na úspěšnost předchozího přenosu. Lze použít jako potvrzovací paket ACK.

POLL – podobný paketu NULL, na rozdíl od tohoto paketu vyžaduje potvrzení od příjemce. Může být užit k volení (poll-ování) zařízení typu slave, který musí odpovídat paketem, i když nemá žádné informace k odeslání.

FSH – speciální řídicí paket obsahující především adresu zařízení a hodinový posun vysílače, nutný pro rychlé spárování obou zařízení. Informační část je plně pokryta 16-ti bitovým CRC kódem. Zabírá pouze jeden časový slot. FSH paket je využíván masterem v části *Page* a *Inguiry* odpovědí a v prohození rolí master - slave. Detailní popis s formátem tohoto paketu je uveden [1].

Datové pakety logické linky ACL

DM1 – délka datové informace je maximálně 18 bajtů. Datová informace je chráněna 16-ti bitovým CRC kódem. Informace včetně CRC jsou kódované (2/3 FEC - Forward Error Correction).

DH1 – podobný typ jako DM1, je schopný přenést až 28 informačních bajtů, ale oproti DM1 není kódován, datová část je opět pokryta CRC kódem. Přenos využívá jeden časový slot.

DM3 – opět paket typu DM1 s rozšířenou datovou částí. Dovoluje přenést až 128 bajtů. Opět je chráněn CRC kódem, přenos obsadí tři časové sloty. Během těchto tří časových slotů nesmí být měněna frekvence.

DH3 – paket typu DM3, který nemá kódovaná data pomocí FEC. Délka dat může být až 185 bajtů plus 16 bitů CRC kódu. Pokrývá tři časové sloty.

DM5 – paket typu DM1 s rozšířenou datovou částí na 226 informačních bajtů, opět pokrytými 16 - ti bitovým CRC kódem. Přenos vyžaduje 5 časových slotů.

DH5 – stejný typ jako DM5, krom toho, že data nejsou kódována pomocí FEC. Pro přenos je opět nutných pět časových slotů. Délka datové části 341 bajtů.

AUX – typ podobný DH1, není chráněn kontrolním součtem CRC a je schopen přenést maximálně 30 bajtů. Pokrývá jeden časový slot.

Shrnutí všech parametrů paketů pro SCO linky je uveden v dodatku B.2 pro ACL linky v dodatku B.3 a pro řídicí pakety v dodatku B.4. Popis kódování FEC a CRC je uveden v [1].

Časování Bluetooth

Každé zařízení Bluetooth obsahuje vnitřní hodiny (Clock), od kterých jsou odvozeny frekvenční skoky vysílače. Jeho hodinový takt je odvozen od volně běžícího nezávislého časovače jehož aktuální hodnotu nelze nijak ovlivnit. Proto se k synchronizaci s ostatními zařízeními používá pouze hodinový offset (Clock Offset). Frekvenční skoky uvnitř *piconetu* jsou určeny hodinami mastera.

Přístupové procedury (Page scan, Page)

Tato část popisuje dva stavy zařízení Bluetooth, které jsou nutné pro navázání spojení. Jde o procedury P*age* a *Page scan*. Aktuální spojení se navazuje pomocí procedury P*age*. Jednotka, která iniciuje spojení (procedurou *Page*) jako první se stává automaticky masterem spojení. Jednotka, která je připravena ke spojení s masterem vyčkává ve stavu *Page scan* se po vytvoření spojení stává slavem. Pro navázání spojení pomocí procedury *Page*, je možné využít několik schémat. Hlavní schéma, které poskytuje každé zařízení se nazývá povinné (mandatory) a je využíváno pro úplně první spojení se vzdálenou jednotkou v čase, kdy nemají zúčastněná zařízení o sobě žádné informace.

Page scan

Je-li jednotka v tomu stavu, vykonává tzv. stránkové scany (Page scan). Jde o funkci, která v určitých časových intervalech (oknech) očekává příjem paketu, ve kterém je umístěn pouze jeho přístupový kód (DAC). Během tohoto časového intervalu jednotka nevyužívá skoků ve frekvenci, ale poslouchá na jedné frekvenci. Interval tohoto okna by měl být minimálně tak dlouhý, aby stačil prohledat 16 stránkovacích frekvencí. Jestli-že jednotka provádějící *Page scan* rozpozná svůj přístupový kód (DAC) vstoupí do stavu *slave response*, který bude popsán níže.

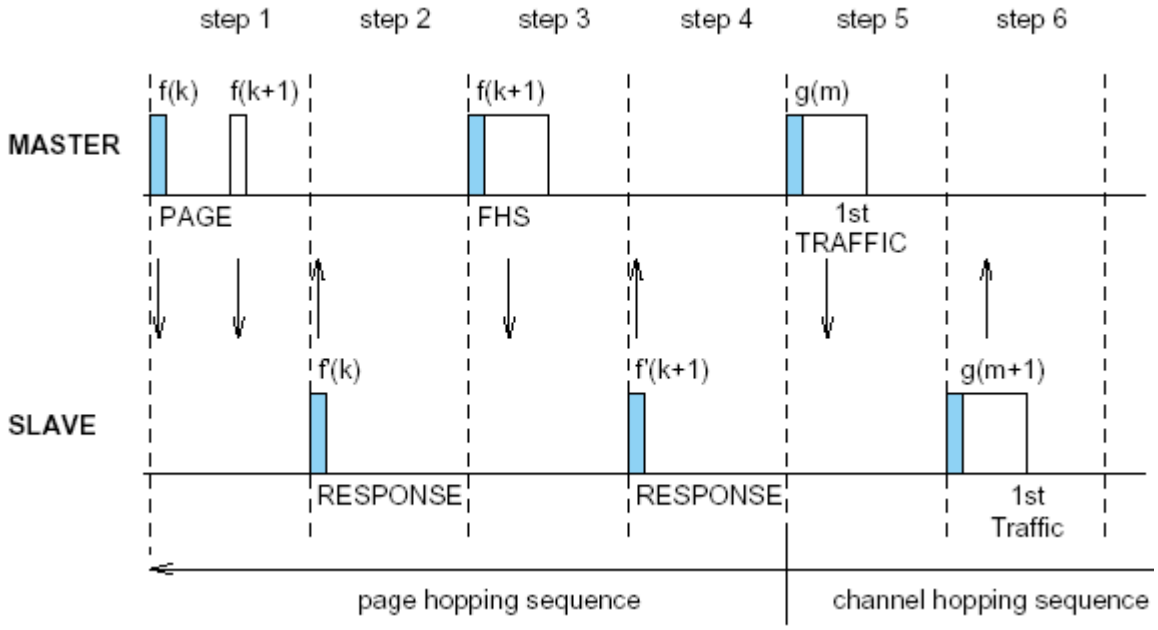
Page

Stav, který užívá zařízení Bluetooth (potencionální master) pro navázání spojení se vzdáleným zařízením (slave), které periodicky provádí *Page scan* (očekává spojení). Master se pokouší získat vzdálené zařízení pomocí vysílání přístupového kódu DAC v různých frekvenčních skocích a mezi vysílacími intervaly poslouchá, zda nepřišla odpověď od vzdáleného zařízení typu slave. Procedura *Page* obsahuje několik kroků. V prvním je nastavena sekvence skoků pode přístupového kódu podřízeného zařízení. Pro určení fáze v sekvenci master odhaduje jeho hodinový offset (buď z posledního navázání spojení nebo pomocí procedury *Inquiry*). Přijme-li zařízení typu slave zprávu od mastera, je zaručena existence spojení mezi těmito jednotkami. Pro nastavení parametru spojení si master a slave vymění důležité informace. Důležité pro komunikaci v *pioconetu* je to, aby měli obě jednotky stejný CAC (channel access code) a užívaly tedy stejnou sekvenci frekvenčních skoků. Jednotka, která iniciovala spojení je automaticky považována jako master. Sekvence skoků kanálu je odvozena od adresy mastera a časování je odvozeno od vnitřních hodin mastera.

Navázaní spojení

Pro navázaní spojení je využíváno výše zmíněných přístupových procedur. Navázaní spojení je rozděleno do několika kroků. Sekvence těchto kroků je vyznačena na obr. 11. Navázání spojení je možné mezi jednotkou ve stavu *Page* (pokouší se o spojení) a jednotkou ve stavu *Page scan* (čeká na spojení). V prvním kroku tedy master pro získání párového zařízení posílá DAC patřící vzdálenému zařízení (slave). Rozpozná-li toto zařízení (slave) v přijaté zprávě svůj vlastní DAC kód, vstupuje do stavu *slave resepons* a odesílá masterovi potvrzení (ID paket). V následujícím kroku master vstupuje do stavu *response master*. Jako odpověď

master odesílá paket FHS s vlastními komunikačními parametry, které budou využity pro nové nastavení parametrů spojení.



Obr. 11 sekvence kroků pro navázání spojení.

Slave response

Je-li jednotka ve stavu *Page scan* pak po přijetí svého vlastního DAC kódu potvrzuje přijetí identifikačním paketem (ID). Tato zpráva obsahuje pouze DAC zařízení slave. Po odeslaní této zprávy (ID) je aktivován přijímač a očekává se příchod FHS paketu. Jestli, že nastane chyba (slave nepřijal FHS paket dříve než vyprší timeout) před vstupem do stavu *Connection* slave se vrací zpět do stavu *Page scan*.

Přijme-li slave FHS paket je následně odeslán paket obsahující DAC jako potvrzení příjmu. Po odeslání tohoto potvrzení změní slave CAC a hodinový offset, které byly přijaty v FHS paketu. V posledním kroku pak vstupuje toto zařízení do stavu *Connection* jako zařízení typu slave. Celý průběh komunikace je uveden na obr. 11.

Master response

V okamžiku, kdy zařízení master přijme odpověď (ID) od protějšího zařízení, vstoupí master do stavu *Master response*. V tomto stavu odešle FHS paket obsahující aktuální stav svého časovače a svou adresu. Po odeslání FHS paketu čeká na další potvrzení (DAC) od párové jednotky. Jestliže není přijato potvrzení, master pošle opět FHS paket s aktualizovanými

údaji. Není-li opět potvrzení přijato do doby než vyprší timeout, vrací se master do stavu *Page* a odešle chybovou zprávu nadřízené vrstvě (Link Manager).

V případě, že potvrzení bylo úspěšně přijato, obnoví master své nastavení časovače a tedy i přístupového kódu kanálu (CAC). V této chvíli jsou nastaveny opět shodné parametry spojení a je možné odesílat data oběma směry.

Procedury pro vyhledávání vzdálených zařízení (Inquiry scan, Inquiry)

Užívá se v situaci, kdy není známa cílová adresa zařízení nebo pro vyhledání dosažitelných jednotek v okolí. Během této procedury se zjišťují adresy vzdálených zařízení a aktuální posun jejich časovače vůči hodinám mastera. Tato dotazovací zpráva je typu broadcast a neobsahuje žádné informace o zdrojovém zařízení. Je možné využít dva dotazovací přístupy, buď GIAC (general inquiry access code) hlavní dotazovací přístup pro jakýkoliv vzdálený modul Bluetooth. Nebo DIAC (Dedicated inquiry access code) pouze pro vyhrazený typ zařízení. Chce-li jednotka objevit ostatní zařízení Bluetooth ve svém dosahu musí vstoupit do stavu *Inquriy*. Ten neustále vysílá *Inquiry* zprávu (ID packet) na různých frekvencích. Jednotka, která má být objevena, pravidelně vstupuje do stavu *Inquiry scan* a odpovídá na zprávu *Inquiry*.

Inqury scan

Stav velice podobný stavu *Page*. Přijímač rozpoznává přístupové kódy tak dlouho, aby kompletně prošel šestnáct dotazovacích frekvencí. Scan je vykonán v jednotlivých skokových frekvencích. *Inquiry* procedura užívá 32 vyhrazených dotazovacích frekvencí podle inquiry sekvencí. Tyto frekvence jsou stanovené GIAC (general inqury adress code) přístupem. Pokud jednotka rozpozná během *Inquiry scanu* zprávu IAC (inquiry access code) vykoná buď návrat do stavu *Connection* nebo *Standby* před pokračováním inquiry scanu, nebo vstoupí do stavu *inquiry response*.

Inquiry

Tento stav využívá jednotka, která chce objevit nové bluetooth zařízení v dosahu. V tomto stavu master obvykle nemá informace o vzdáleném zařízení. V této proceduře tedy odešle ve dvou následných skokových frekvencí paket IAC. Mezi vysláním dotazu inquiry, přijímá jestli nebyla obdržena zpráva (FHS paket) od jakéhokoliv vzdáleného zařízení. Po přijetí této

zprávy zařízení master pokračuje ve vysílání dotazu na zařízení do té doby dokud není ukončen pomocí Link Manageru (zodpovídá za správu spojení) Poté může toto zařízení přejít do stavu *Connect* (využitím stavu *Page*) nebo *Standby*.

Inquiry response

Na operaci *Inquiry* může reagovat pouze zařízení, které nebude v konečném spojení nastaveno jako master, existuje tedy jen jeden možný stav nazvaný *slave response*. Master sice ve stavu *Inquiry* přijímá, zda nepřišla odpověď od vzdáleného zařízení (FHS paket), ale příjem tohoto paketu nijak neovlivní stav mastera. Pokračuje tedy stále v prohledávání prostotu. Jestliže zařízením typu slave je ve stavu I*nquiry* scan a přijme paket typu IAC, musí odeslat informační paket FHS (obsahující vlastní adresu a aktuální hodinový cyklus).

Jestliže je v blízkém okolí několik jednotek Bluetooth a všechny budou odpovídat ve stejném čase, mohl by vzniknout problém a dojít ke kolizi. Jelikož má každá jednotka volně běžící hodiny, proto je velice nepravděpodobné, že všechny užijí stejné fáze v sekvenci změn frekvence (inguiry hopping sequence). Za účelem odstranění kolizí mezi jednotkami Bluetooth je implementován následující protokol. Jestliže slave přijme dotaz inquiry (IAC paket), vygeneruje se náhodné číslo mezi 0 – 1023. Slave se pak vrátí v časovém slotu, který je určen náhodně vygenerovaným číslem, do stavu *Connection* nebo *Stanby*. Slave pak odesílá paket FHS až v dalším přijetí dotazu inguiry (IAC paket).

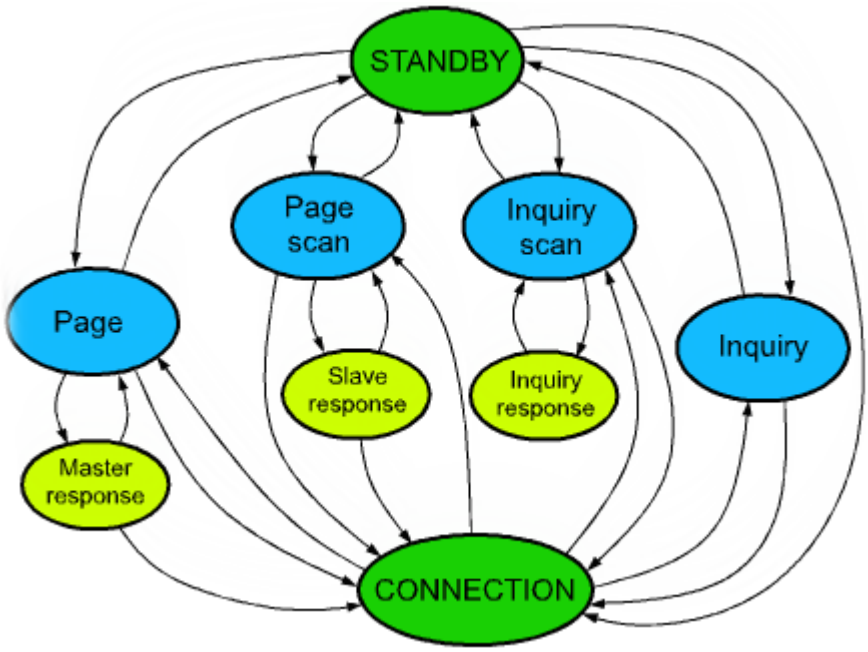
Stav Connection

Ve stavu *Connection* je navázáno spojení a pakety se mohou přenášet oběma směry. Obě jednotky využívají přístupový kód kanálu odvozený od mastera. Sekvence frekvenčních skoků nastavena také podle mastera. Přenos začíná master v sudem slotu slave v slotu lichém.

Stav Standby

Je o výchozí stav jednotky Bluetooth. V tomto stavu má jednotka malý odběr. Běží pouze vnitřní hodiny.

Stavový diagram s možnými přechody mezi jednotlivými stavy komunikace je uveden na obr.12.

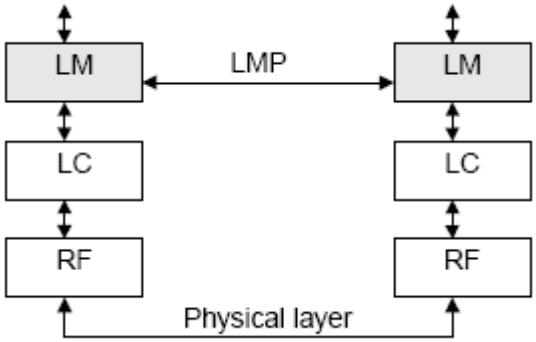


Obr. 12 diagram přechodů mezi jednotlivými stavy (Link Controller).

2.2.3 Protokol linkového manageru (LMP)

Protokol linkového manageru (Link Manager Protokol – LMP) je užíván k nastavení, zabezpečení a řízení komunikačních linek. Dále pak synchronizaci při vytváření spojení pomocí linek ACL a SCO, šifrování spojení, detekci chyb a korekci chyb. Umožňuje ověření (autorizaci) při navázání spojení, pomocí privátních klíčů s využitím schématu výzva – odpověď (challenge - response) [1].

Zprávy manageru jsou zachyceny opět linkovým managerem a nejsou propuštěny na vrstvy vyšší (obr.13). Zprávy linkového manageru mají vyšší prioritu než data.

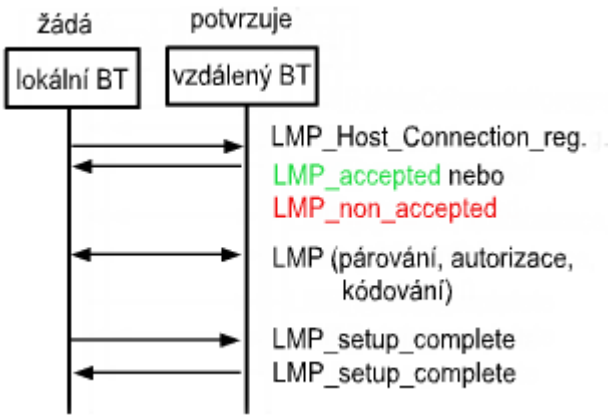


Obr. 13 komunikace na úrovní linkového manageru.

Jelikož v této práci nebylo využito autorizace spojení ani kódování dat, není nutné je zde uvádět a detailní popis lze nalézt v [1] nebo v [2].

Navázání spojení na úrovni linkového manageru

Po navázání komunikace mezi moduly Bluetooth, pomocí procedury *Page* dochází k cyklickému dotazování na úrovní *LM* (Linkový Manager). Pro dotazování je využito několik procedur, které poskytnou informace o vzdáleném zařízení, možnosti kódování a autorizace. Jsou-li všechny předchozí akce úspěšné, dojde k vytvoření spojení. Chce-li jednotka master navázat spojení odešle požadavek na spojení (LMP_host_connection_req). Poté je očekáváno zda vzdálené zařízení pošle potvrzení o akceptovaní (LMP_accepted) spojení či nikoliv (LMP_not_accepted). Je-li spojení akceptováno uplatní se procedury pro párování (Pairig), ověření pravosti (authentication) a kódování (encryption). Není-li zařízení po uplatnění všech těchto bezpečnostních procedur v počátečním stavu (např. došlo k chybě při ověření) je odeslána zpráva o kompletním navázaní spojení (LMP_setup_complete). Celý průběh žádostí a odpovědí je uveden na obr. 14.



Obr. 14 navázání spojení pomocí LMP.

2.2.4 Vrstva HCI (Host Controller Interfaces)

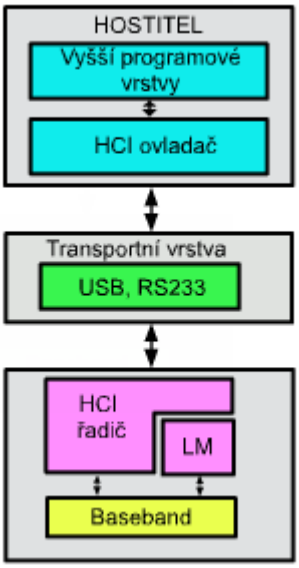
Vrstva HCI poskytuje jednoduché rozhraní mezi jádrem Bluetooth a aplikační částí. Zajišťuje jednotnou metodu přístupu k nižším vrstvám a k celému hardwaru jednotky Bluetooth. Jde tedy o vrstvu, která poskytuje příkazové rozhraní pro správu kanálu, monitorování stavu hardwaru, řídicích registrů a registrů událostí. Vrstva HCI se skládá ze tří

sekcí hostitel (Host), transportní vrstva (Transport Layer) a řadič hostitele (Host Controller). Seskupení těchto sekcí je na obr.15.

- Hostitel (Host) – jde o část ovladače HCI (HCI driver) a tedy o software, který na základě HCI příkazů a událostí, zprostředkovává služby modulu Bluetooth vyšším aplikačním vrstvám. Je to část HCI, která vytváří spojení s vyššími programovými vrstvami.

- Transportní vrstva (Transport Layer) – je komunikační vrstva zajišťující spojení ovladače HCI (HCI driver) a řadičem HCI (Host controlle). Tato vrstva poskytuje několik možných komunikačních rozhraní, kterými jsou vybaveny běžné osobní počítače. Jde o rozhraní USB (Universal Seriál Bus) nebo RS232, popřípadě UART.

- Řadič HCI (Host Controller) – ve této části jsou implementovány příkazy HCI, které tento řadič vykonává prostřednictvím nižších vrstev jádra Bluetooth (baseband, Link controller).



Obr. 15 propojení jednotlivých vrstev HCI.

Jak již bylo výše uvedeno, vrstva HCI využívá k přístupu na nižší vrstvy protokolu tzv. HCI příkazů. Jelikož provedení těchto příkazu může být různě časově náročné, jsou hostiteli (Host) výsledky těchto příkazů oznamovány pomocí událostí (Events). Pakety definované pro vrstvu HCI se dělí na čtyři typy (tab.3). Tento typ je součástí každého paketu a je odeslán jako první, pak následuje příslušný paket.

Typ HCI paketu	HCI identifikator
HCI příkaz	0x01
HCI ACL datový paket	0x02
HCI SCO datový paket	0x03
HCI událost	0x04

Tab. 3 typy HCI paketů.

HCI příkazy

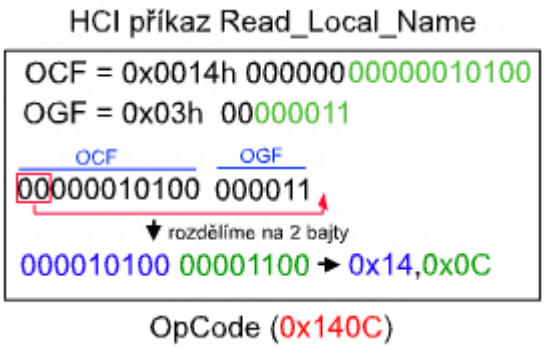
V této části budou popsány typy a konstrukce HCI příkazů. Jestliže nebude uvedeno jinak, jsou všechny hodnoty parametrů zapsány, odeslány a přijaty binárně či hexadecimálně a to v *Little endian* formátu. Tohoto formátu využívají i parametry HCI příkazu (pro více bajtové parametry je nejméně významný bajt umístěn nejvíce vpravo a je přenesen jako první). Na parametry typu *string* (např. názvy) se tyto pravidla nevztahují. HCI příkazy se dělí na šest skupin. Označení příslušné skupiny, do které příkaz spadá, uvádí část OGF (OpCode Group Filed). Bajtové pole označené OCF (OpCode Command Filed) označuje konkrétní příkaz v dané skupině. Jednotlivé skupiny HCI příkazu jsou uvedeny v tab. 4. Formát paketu HCI příkazu je uveden na obr. 16. Před tímto paketem je umístěn typ tohoto paketu, který je podle tab.3 roven **0x01**.

OGF příkazu	Group	popis
0x01	Link Control Commands	Řídicí příkazy pro spojení
0x02	Link Policy Commands	Příkazy pro správu jednotek
0x03	Host Controller & Basebad Commands	Řízení a nastavení Bluetooth
0x04	Informational Parameters	Informační parametry
0x05	Status Parameters	Stavové parametry
0x06	Testings Commands	Testovací příkazy

Tab. 4 skupiny HCI příkazů.

Obr. 16 formát paketu HCI příkazu.

Tento paket je složen z několika částí. První část *OpCode* je složen ze dvou polí (OCF a OGF). Složením těchto dvou částí vytváří tzv. *OpCode*. Příklad složení *OpCodu* je uveden na obr. 17.



Obr. 17 vytvoření *OpCodu* HCI příkazu.

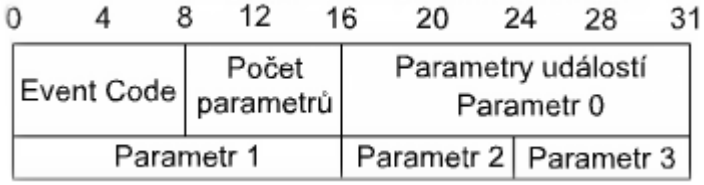
Dalším atributem HCI příkazu je celková délka přenášených parametrů (v bajtech). Ten zabírá pouze jeden bajt. Dále už následují konkrétní parametry jejichž počet a délka závisí na typu příkazu.

HCI události

Tato část se zabývá popisem a formátem HCI událostí, které jsou užity vrstvou HCI jako odpověď na HCI příkazy. Každý HCI příkaz nemusí nutně očekávat pouze jednu událost. Počet a typ událostí k jednotlivým příkazům je popsán v [1].

Jde o rozsáhlou skupinu událostí, které nejsou žádným způsobem rozděleny. Výjimku tvoří události jejichž kód je 0xFF a 0xFE, které jsou rezervovány pro ladění a testování. Formát paketu událostí je uveden na obr. 18. První částí tohoto paketu je kódové označení příslušné události. Následný parametr určuje celkový počet všech parametrů, které tato

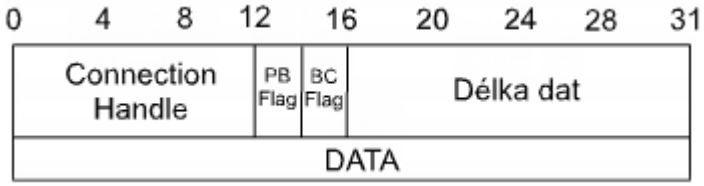
událost obsahuje. Dále už následují pouze konkrétní parametry. Před tímto paketem je opět umístěn typ a podle tab. 3. je roven **0x04**.



Obr. 18 formát paktu HCI událostí.

HCI datový paket (ACL)

Tento typ paketu je užíván pro výměnu dat mezi HCI ovladačem (Host) a HCI řadičem (HCI Controller). Formát datového paketu je uveden na obr. 19. Před ním je opět uveden typ paketu podle tab. 3 a je roven **0x02**.



Obr. 19 formát datového paketu.

První část tohoto paketu je číslo datového spojení (*Connection Handle*), které je získáno jako parametr obsažený v události na HCI příkaz, pro vytvoření datového spojení (Create_Connection). *PB Flag* (Packet Boundary Flag) je využíván vrstvou L2CAP a označuje zda daný paket je první (nastavení bitů 10b) nebo zda jde o pokračování rozděleného paketu (nastavení 01b). Následující dvojice bitů je označena jako *BC Flag* (BroadCast Flag), který označuje jestli je daný datový paket určen jen pro spojení mezi dvěma zařízeními (nastavení 00b) nebo zda je určen pro všechna aktivní zařízení v *piconetu* (nastavení 01b nastavení 10b – i pro zařízení v *Park* módu). Poslední částí datového paketu je celkové délka přenášených dat. Bližší informace jsou uvedeny v [1].

2.2.5 Vyšší vrstvy zásobníku

Jsou to vrstvy umístěné nad vrstvou baseband, zajišťující širší využití protokolu Bluetooth.

L2CAP (Logical Link Control and Adaptation Protokol) - tato část specifikaci Bluetooth je umístěna nad protokolem baseband (obr.1) a využívá jeho fyzické datové linky ACL, linka SCO podporována není. L2CAP je tedy určena pro vyšší vrstvy, kterým poskytuje spojové (Connection-Oriented) služby a služby bez spojení (ConnectionLess - Oriented), detailněji v [1]. Obecně zastává tyto úlohy:

- Multiplexování (Multiplexnig): podporuje přepínání několika druhů protokolů, které jsou definovány vyššími vrstvami protokolového zásobníku Bluetooth. Jako je např. SDP, RFCOMM a TCS – BIN, které budou popsány níže.

- Segmentace a spojování (Segmentation and reassembly): část této vrstvy umožňuje přenos datových paketů, jejichž délka přesahuje maximální možnou délku, kterou je schopna přenést vrstva baseband. Je tedy zodpovědná ze rozdělení (vysílací strana) a opětovné složení (přijímací strana) datové částí paketu. Podporuje délku paketu až 64 *kB*.

- Kvalita služeb (Quatlity of Services): proces navázání spojení povoluje výměnu informací s ohledem na kvalitu služeb mezi dvěma jednotkami. Musí zajistit předem definované a dohodnuté parametry.

- Skupiny (Groups): implementace práv pro mapování skupin jednotek do sítě *piconet*.

SDP (Service Discovery Protokol) – definuje, jakým způsobem klientská jednotka Bluetooth hledá využitelné služby serverů Bluetooth. Tento protokol tedy slouží k výměně informací o službách dostupných na jednotlivých zařízeních. Detailní informace jsou uvedeny v [1].

RFCOMM (Radio Frequency Communications port) – tento protokol slouží jako emulace sériového protokolu, je vybaven emulací protokolu RS232 (EIATIA – 232 - E). Lze tedy s výhodou použít pro aplikace využívající pro přenos dat klasický sériový port.

TCS – BIN (Telephony control - Binary) – jde o bitově orientovaný protokol, definující řízení, sestavení přenosové linky a přenos hlasu a dat mezi jednotkami Bluetooth.

3. Kapitola

Popis modelu a návrh hardwarových částí

V úvodu této části je uveden popis reálného modelu vzducholodi a jeho periferních zařízení pro orientaci v neznámém prostoru. Následující část této kapitoly se zabývá konkrétním řešením zadaného problému v části návrhu příslušného hardwaru. Oblasti jeho softwarové podpory a popis programového vybavení pro obě strany komunikace bude uveden v následující kapitole (4).

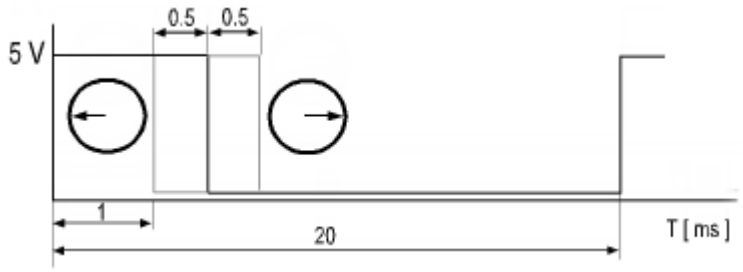
3.1 Model vzducholodi

Jde o funkční model (obr.20) jehož náplň tvoří inertní plyn hélium. Pro pohyb v prostoru je model vybaven třemi výkonnými modelářskými DC motory pro pohyb v horizontálním a vertikálním směru a směrový motor pro otáčení modelu ve směru horizontálním. Na spodní části modelu je gondola, ve které je umístěna řídicí a komunikační elektronika a zdroj napětí. Model je vyvažován olověnými závažími. Nosnost tohoto modelu je dimenzována na 0.5 *kg* včetně všech periferních zařízení.

Obr. 20 model vzducholodi.

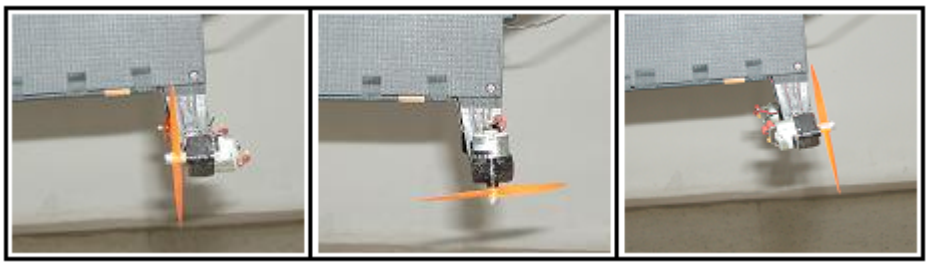
3.1.1 Jednotky pohonů modelu

Ovládání tohoto modelu v prostoru zajišťují tři stejnosměrné modelářské motory. Dva jsou umístěny vespod modelu na otáčivé ose, umožňující natočení těchto motorů do polohy kolmé k modelu pro udržení výšky a do polohy vodorovné k modelu pro určení směru pohybu. Natočení osy hlavních motorů je realizováno modelářským polohovým servopohonem, proto je přechod mezi těmito krajními polohami spojitý. Možné natočení je uvedeno na obr.22. Řízení polohových servopohonů je realizováno pomocí PWM modulace s modulačním plněním jak je uvedeno na obr.21. Frekvence impulsů je 20 *Hz* a určuje jakou rychlostí se budou servopohony přestavovat.



Obr. 21 řízení polohového servopohonu pomocí PWM.

Stejnosměrné motory jsou řízeny také PWM modulací, ale se odlišným plněním (0 – 70%). Pro zachování standardního řízení uvedeného na obr.21 pro všechny pohony, je nutné předřadit rychlostním pohonům dekodér PWM s řízením směru otáček. Jde o součást, která byla dodána s modelem a není k ní bližší popis. Obsahuje integrovaný zákaznický obvod HT700 od firmy Hitec, jehož výstup řídí čtyři výkonové tranzistory zapojené v H – můstku [15].



Obr. 22 možnosti natočení hlavního pohonu.

3.1.2 Zařízení pro orientaci v prostoru

Pro orientaci v prostředí s překážkami byl model doplněn o čtyři ultrazvukové dálkoměry s dosahem cca 15 *m*, jejichž konstrukce je náplní paralelně realizované diplomové práce [3]. Dále model disponuje dvouosým elektronickým kompasem *PNIV2Xe*, který měří magnetické pole země. Jde o průmyslový modul firmy PNI komunikující s nadřazeným zařízením pomocí sériové synchronní sběrnice SPI. Detailní informace jsou uvedeny v [9]. Pro správné určení azimutu vyžaduje elektronický kompas umístění ve vodorovné poloze.

Aktuální polohu modelu lze určit jen v ohraničeném prostoru. A za podmínky, že vzdálenost hranic od modelu je možné vždy určit (dosah ultrazvukových dálkoměrů). Jelikož model není vybaven systémem pro globální navigaci (GPS), není možné určovat polohu modelu ve volném prostoru.

3.1.3 Řídicí jednotka modelu

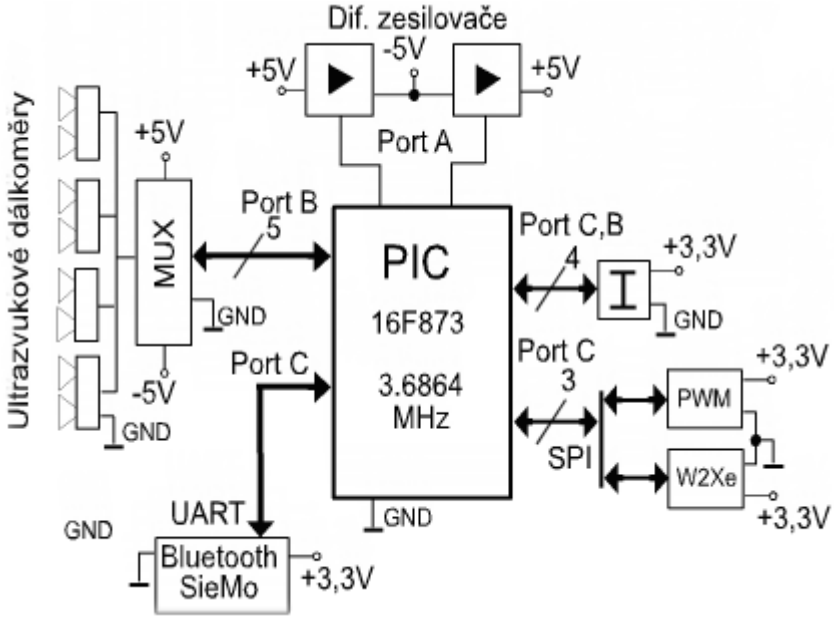
Model obsahuje mnoho periférních zařízení, které je nutné nějakým způsobem řídit a získat tak potřebná data. Proto je model doplněn o řídicí jednotku, jejímž jádrem je jednočipový mikrokontrolér PIC 16F873 od firmy Microchip pracující s taktovací frekvencí 3,6864MHz.

K tomuto mikrokontroléru je připojena řada periférií, které ještě nebyly výše zmíněny. Jde o přístrojové zesilovače pro měření odběru a napětí baterií. Další dohlížecí obvody pro kontrolu maximálního odběru, nabíjecích proudů a změny napětí na bateriích.

Připojení ultrazvukových dálkoměrů zajišťuje mulitplexer. Ten na vstup mikrokontroléru připojí v jedné chvíli pouze jeden dálkoměr. Volba dálkoměru je učena adresou na vstupu multiplexeru. Je tedy možné v jedné chvíli měřit jen jedním dálkoměrem. To je nutné i z hlediska možných přeslechů u jednotlivých ultrazvukových dálkoměrů.

Pro řízení modelářských servopohonů byl mikrokontrolér vybaven PWM modulátorem, který je tvořen jednoduchým mikrokontrolérem PIC 16F819. Řídicí jednotka tedy obsahuje dva mikrokontroléry, které pro vzájemnou komunikaci využívají synchronní sériové sběrnice SPI. Mikrokontrolér zastávající funkci PWM modulátoru je vždy jako slave a přijímá příkazy od řídicího mikrokontroléru. V závislosti na těchto příkazech mění hodnoty plnění PWM. Na obr.23 je tento mikrokontrolér znázorněn jako blok PWM.

Rozhraní SPI využívá i elektronický kompas V2Xe, který má třívoltové rozhraní, a proto musí být oddělen pomocí budiče sběrnice SPI jehož výstupní napětí je v rozsahu této logiky. Posledním zařízením připojeným k mikrokontroléru je vestavný modul SieMoS50037, jehož podrobný popis a řízení bude uvedeno v následující části této kapitoly. Tento modul podporuje rozsah komunikační signálu od 0 do 3.3 *V*. Na obr.23 je orientační blokové schéma řídicí části modelu. Detailní informace jsou uvedeny v [3], tato část je předmětem diplomové práce [3]. Celá řídicí část je napájena ze zdroje napětí, který tvoří velmi lehké modelářské lithium-polymerový akumulátor, které jsou zapojeny do série.Výstupní napětí tohoto napájecího bloku je 8.4 *V*. Všechna zařízení umístěná v modelu ovšem nemají shodné napájení, proto musí být celý systém vybaven také částí pro úpravu napájecího napětí. Změna napájecího napětí je zajištěna pomocí DC/DC měničů od firmy Maxim. Tyto integrované obvody jsou součástí zapojení pro změnu a stabilizaci tohoto napětí. Pomocí těchto obvodů lze dosáhnout i napětí, kterým nedisponuje zdrojový akumulátor. Přesný popis zapojení je uveden v [3], kde lez nalézt všechny příslušné detaily.



Obr. 23 blokové schéma řídicí části modelu.

3.2 Popis modulu SieMo – S50037

Hlavní součástí bezdrátové komunikace je vestavý modul Bluetooth třídy 2 (Class 2) od firmy Siemens. Jde o modul, který může být umístěn do různých elektronických zařízení, vyžadující bezdrátové spojení. V tomto případě byl modul SieMo umístěn do reálného modelu vzducholodi. Druhá strana komunikace je zajištěna opět tímto modulem, který je připojen k osobnímu počítači (obr.26).

3.2.1 Vlastnosti modulu SieMo

Vestavný modul *SieMo S50037* (obr.24) je modul třídy 2 (Class 2). Jde tedy o zařízení s dosahem do 20 *m*. V jádru je zde implementován protokol Bluetooth verze 1.1 z ledna roku 2001. Jak již bylo uvedeno výše jde o protokol, který byl v současné době nahrazen novější verzí. U nové verze by měla být zachována zpětná kompatibilita. Pro připojení k vnějším obvodům disponuje modul několika rozhraními. Z pohledu užití toho modulu jako vestavného (embedded) zařízení, jsou důležitá převážně rozhraní UART a SPI. Pro přenos dat je u tohoto modulu možné využít dvou přenosů. Buď synchronní přenos (64 *kb/s*) nebo asynchronní datový přenos (symetrický 432,6 *kb/s* nebo asymetrický 723 *kb/s* v jednom směru a 57,6 *kb/s* v opačném směru). Bezdrátový přenos probíhá na frekvenci 2.4 *GHz* s rozsahem od 2,402 až 2,480 *GHz*. Obsahuje 79 kanálů se šířkou měnící se po 1 *MHz*, využívající pro přenos pásmo ISM (Industrial Scientific Medical band, bezlicenční radiové pásmo) Nosný signál je modulován pomocí GFSK (Gaussian Frequency Shift Keying) modulací.

Souhrn významných parametrů:

- Podporuje protokol Bluetooth v 1.1.
- Nejnižší programově přístupná vrstva firmware – HCI (Host Command Interface).
- Bluetooth zařízení třídy 2 (vzdálenost mezi zařízeními do 20 *m*).
- HCI přístupné přes UART, BCSP (BlueCore Serial Protocol [5]) a USB.
- Rychlost datového přenosu přes UART až 1,5 *Mb/s*, USB (1.1) high speed.
- Frekvenční rozsah radiového signálu 2.402-2.480 *GHz.*
- Napájení 3.3 *V*, max. 150 *mA*.
- Typ jádra Bluetooth modulu BC01 (CSR Bluetooth chip).

3.2.2 Rozhraní modulu SieMo S50037

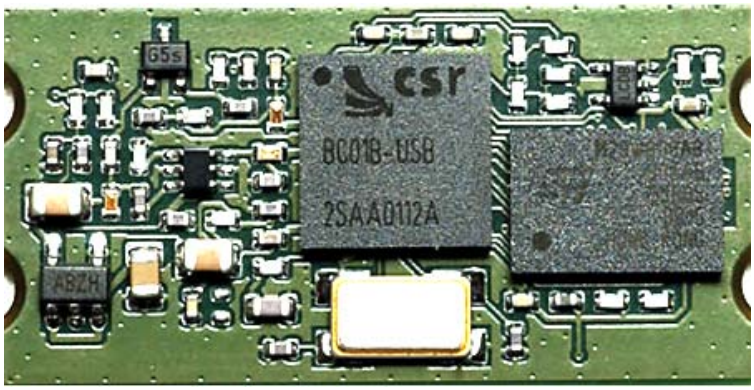
Tento modul obsahuje několik rozhraní pro připojení k vnějším řídicím obvodům či přímo k osobnímu počítači.

PCM: pulsně kódová modulace, je jedna z metod pro digitalizaci zvuku a přenos přes digitální kanál. Poskytuje jak příjem tak vysílání dat pomocí SCO. Modul může pracovat v režimu buď jako master (s výstupní frekvencí 256 *kHz*) nebo jako slave (se vstupní frekvencí v rozmezí 128-512 *kHz*). Master mód je nastaven jako standardní. Slave mód je třeba nastavit v primární konfigurační databázi (Persistent Store Key). Detailní informace lze nalézt v [4].

UART: Univerzální asynchronní přijímač vysílač, obsahuje standardní UART 16550 kompatibilní rozhraní. Standardní rychlost komunikace od 9600 *Bd* až 1.5 *MBd*. Komunikační rozhraní nevyžaduje nutně řízení hardwarového toku dat a je možné jej vypnout v primární konfigurační databázi. Standardně je hardwarové řízení toku dat povoleno a vyžaduje zapojení nejen linek Rx a Tx, ale také linky RTS a CTS. Pro připojení k osobnímu počítači je nutná úprava úrovní signálů (obr. 26). Pomocí toho rozhraní lze přistoupit přímo na vrstvu HCI. Toto rozhraní poskytuje přístup jak pro čtení tak pro zápis do primární konfigurační databáze modulu.

USB: universální sériová sběrnice,podporuje protokol USB 1.1. Jádro BC01 má integrován radič USB typu slave, musí být tedy částí hierarchie obsahující alespoň jednoho mastera. Obsahuje vlastní vysokorychlostní rozhaní (full-speed). Pomocí tohoto rozhraní lze také přistoupit na vrstvu HCI. Pro toto rozhraní jsou výrobcem dodávány originální ovladače.

SPI: sériové synchronní rozhraní využívající SPI standard od firmy Motorola. Modul je vždy v pozici slave, SPI_CLK je tedy nastaveno zařízením typu master (např. PC). Modul využívá během přenosu 16 - ti bitovou adresu a 16 bitů dat. Aby mohl být modul úspěšně adresován a využit tak pro komunikaci, je třeba nastavit SPI_CSB (chip select) na nízkou úroveň. Data jsou časována na náběžnou hranu hodin (SPI_CLK). Podrobnější informace jsou opět uvedeny v [4].

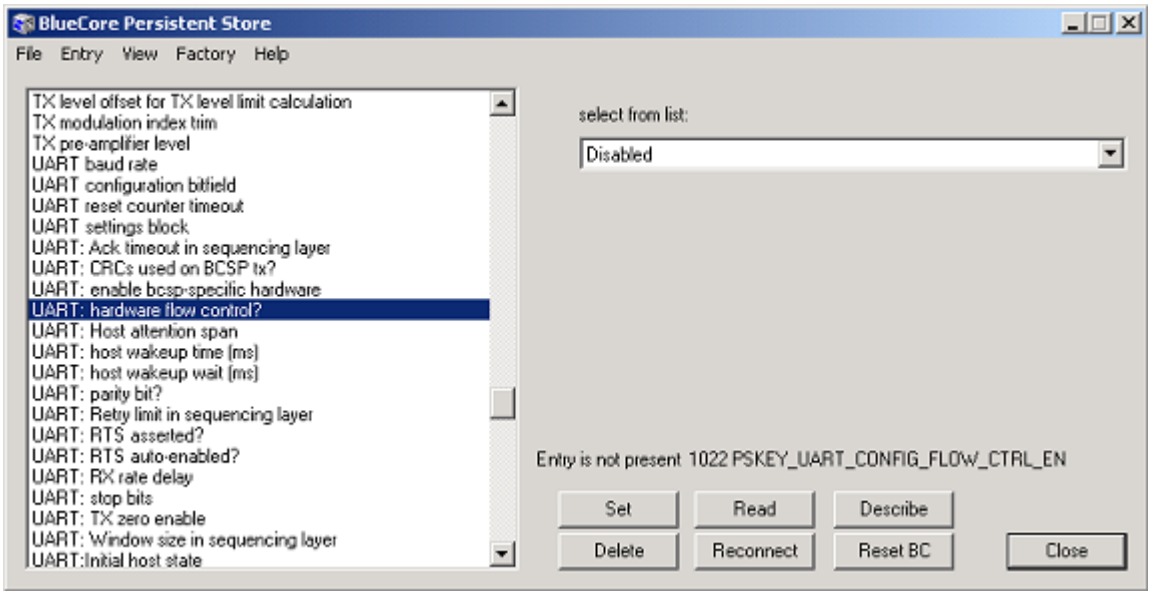


Obr. 24 modul SieMo S50037.

3.2.3 Primární konfigurační databáze (Persistent Store Key)

Pod tímto názvem se skrývá velice široká skupina parametrů Bluetooth modulu SieMo S50037, které jsou využity pro optimální nastavení chování modulu. Tyto parametry jsou uloženy v paměti typu flash a jsou zachovány i po vynutí modulu. Veškeré parametry uvedené v této databázi jsou nastaveny výrobcem. Jelikož jde o vestavné moduly, je možné některé parametry změnit podle potřeby aplikace, která tyto Bluetooth moduly využívá. Celý soubor těchto klíčů je rozčleněn do třech skupin. První skupina parametrů je nastavena jako výchozí a předpokládá se jejich změna. Druhá skupina parametrů je závislá na specifických vlastnostech konečného návrhu a neměla by být měněna. Třetí skupinou jsou parametry, které nesmí být měněny za žádných okolností. Celkový počet parametrů přesáhne 300. Z oblasti klíčů, které jsou povoleny měnit, jde převážně o nastavení parametrů jednotlivých komunikačních portů, připojení externího krystalu, konfigurace 24 bitů BD adresy modulu, třídu zařízení nebo třeba kód země. Parametry, které nejsou určeny ke změnám zde nebudou uvedeny. Bližší popis těchto parametrů lze nalézt v [6].

Pro změnu těchto parametrů je lze využít některého z výše popsaných rozhraní. Změnu těchto parametrů je možné provést pomocí programu dodávaného výrobcem (např. programový balík BlueSuite obsahující program PStool obr. 25).



Obr. 25 program PStool pro konfiguraci primárních parametrů.

3.3 Návrh hardwarové části

Podle zadání byl pro bezdrátovou komunikaci zvolen modul SieMo S50037. Jde o vestavný modul, u kterého se předpokládá, že bude představovat součást nějakého jiného zařízení, kterému poskytne možnost bezdrátové komunikace s okolím. Tento modul tedy není navržen tak, aby bylo možné ho přímo připojit k osobnímu počítači přes sériové rozhraní.

V prvé části práce bylo tedy nezbytné zajistit připojení modulu SieMo k osobnímu počítači, vytvořit bezdrátové spojení mezi osobními počítači a ověřit tak správnou funkci modulu SieMo a protokolu Bluetooth.

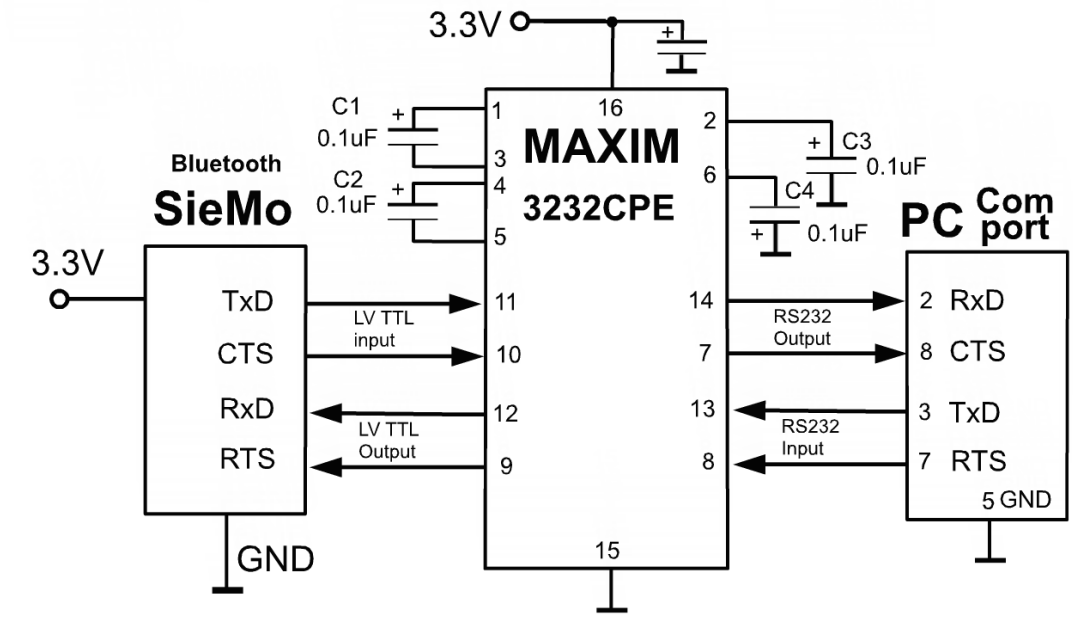
Druhá část této práce se zabývala náhradou osobního počítače jednočipovým mikrokontrolérem PIC, v kterém byla implementována část protokolu Bluetooth, pro komunikaci s řídícím počítačem. Kromě tohoto protokolu bude mikrokontrolér obsahovat řídicí program pro ovládání senzorů a pohonů modelu.

V následující části bude popsán způsob připojení toho modulu jak k osobnímu počítači, tak ke zvolenému mikrokontroléru PIC 16F873, který je umístěn v reálném modelu.

3.3.1 Připojení modulu SieMo k osobnímu počítači

Přístup k jádru *BC01* modulu *SieMo* byl zadán přes vrstvu HCI. Ta je zároveň nejvyšší vrstvou implementovanou v jádru. Přístup na tuto vrstvu poskytuje maximální využití a konfiguraci modulu Bluetooth.

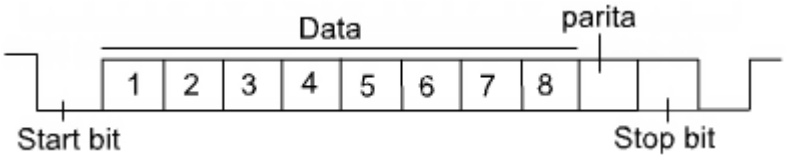
Jak již bylo zmíněno v popisu modulu SieMo, lze na HCI vrstvu přistoupit pomocí rozhraní UART, které lze pomocí externích obvodů přizpůsobit tak, aby bylo možné ho připojit k osobnímu počítači. Vzhledem k tomu, že jde pouze o modul, nedisponuje přímo signály podle RS232C, které obsahuje každý osobní počítač. Modul využívá třívoltové logiky a je tedy nezbytné doplnit modul o externí napěťový převodník úrovní. Schéma zapojení je podle [4] na obr. 26. Lze využít napěťový převodník od firmy Maxim (MAX 3232CPE). Ten zajistí bezpečný převod napěťových úrovní v obou směrech. Tento obvod využívá externích kondenzátorů a pracuje na principu dvojité nábojové pumpy. Poskytuje tedy rozsah úrovní +/- 5.5 V.



Obr. 26 schéma zapojení Bluetooth modulu k osobnímu počítači.

Mezi důležité části patří nastavení komunikačních parametrů obou zařízení. Je třeba nastavit stejné komunikační parametry pro oba účastníky komunikace. Modul Bluetooth je standardně nastaven na komunikační rychlost 115.2 *kBd* a hardwarové řízení toku dat (signály RTS, CTS). Rámec je ukončen jedním stop bitem a neobsahuje paritní bit [4]. Tyto

parametry jsou pro moderní osobní počítač standardní a lze je bez problémů nastavit. Grafické znázornění komunikačního rámce je na obr. 27.



Obr. 27 grafická reprezentace rámce pro sériovou komunikaci.

3.3.2 Připojení modulu SieMo k mikrokontroléru PIC

Pro komunikaci s bluetooth modulem, který bude umístěn v reálném modelu, je využito jednočipového mikrokontroléru PIC 16F873 firmy Microchip. Tento mikrokontrolér poskytuje všechny potřebné periférie. Je také vybaven sériovým komunikačním rozhraním UART, ale opět zde nejsou shodné napěťové úrovně signálu. Na rozdíl od spojení Bluetooth modulu s osobním počítačem je třeba napěťově upravit rozsah signálů pouze směrem od mikrokontroléru k Bluetooth modulu (linky TxD a CTS), neboť výstup modulu je v úrovních LV TTL, které vyhovují rozsahům vstupních pinů mikrokontroléru. Opačný směr (linky RxD RTS) nelze přímo propojit, protože Bluetooth modul pracuje pouze na úrovni třívoltové logiky. V následující tabulce je shrnut rozsah napěťových úrovní jednotlivých součástí.

Zařízení	log.0	log.1
BT modul	0 V	3.3 V
PIC 16F873	0 V	5 V
PC COM	+6 V	-6 V
PC LPT	0 V	5 V

Tab. 5 napěťové úrovně jednotlivých zařízení.

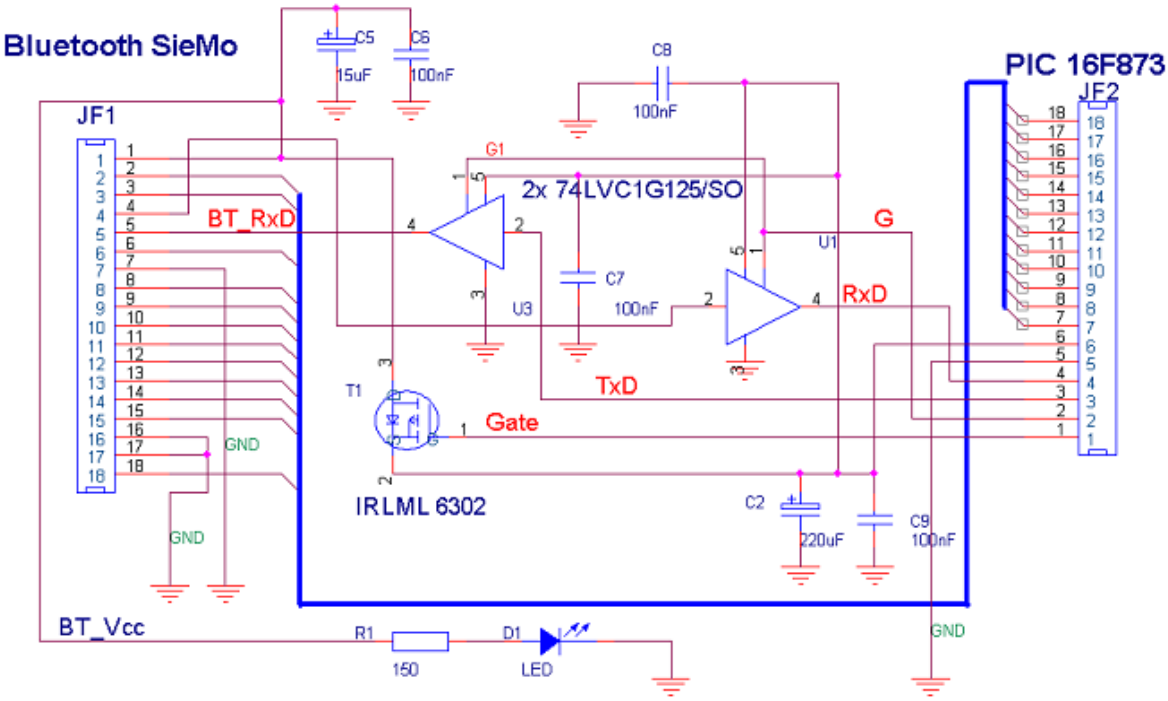
V části nastavení komunikačních parametrů rozhraní UART je oproti osobnímu počítači situace odlišná. Jak již bylo uvedeno je modul Bluetooth standardně nastaven na komunikační rychlost 115.2 *kBd* a hardwarové řízení toku dat. Ovšem oba tyto parametry není možné podle [10] nastavit v mikrokontroléru. Ten podporuje maximální rychlost komunikace 57,6 *kBd* (odvozena od hodinového taktu procesoru) a řízení toku dat nepodporuje vůbec. Softwarová emulace řídicích signálů je zbytečná, protože komunikace

s modulem Bluetooth ji striktně nevyžaduje. Nejschůdnější cesta jak docílit stejných parametrů komunikace je změnit tovární nastavení modulu SieMo v oblasti primární databáze. Změna byla provedena pomocí originálního programu PStool od firmy CSR (obr.25) a modul byl připojen k osobnímu počítači podle obr. 26. Výsledné komunikační parametry obou modulů jsou uvedeny v tab. 6.

Připojení k	rychlost [kBd]	řízení toku	stop bit	parita	označení
PC	115.2	(CTS,RTS)	1	žádná	modul 1
PIC 16F873	57.6	žádné	1	žádná	modul2

Tab. 6 nastavení komunikačních parametrů rozhraní UART.

Příklad připojení modulu Bluetooth k řídicímu mikrokontroléru je uveden na obr. 28. Připojení je realizováno pomocí budících integrovaných obvodů 74LVC1G125 od firmy Texas Instruments, pro úpravu napěťových úrovní jednotlivých signálů. V tomto zapojení je konvertor úrovní vložen do obou signálových cest (RxD, TxD), z důvodů možného odpojení obou signálových cest pomocí řídicího vstupu G. Tak je možné modul Bluetooth programově odpojit a připojit. To je s výhodou využito při inicializaci procesoru. Pro kontrolu napájení je obvod doplněn o SMD LED diodu. Podrobný popis zapojení řídicího modulu umístěného v modelu je uveden v [3].



Obr. 28 připojení modulu Bluetooth k řídicímu mikrokontroléru.

3.3.3 Vývojové zařízení pro modul Bluetooth

Tento přípravek byl navržen a postaven jako vývojový prostředek pro tvorbu nejen počítačového softwaru, ale i pro vývoj řídicího programu pro mikrokontrolér. Návrh byl veden také k tomu, aby bylo možné využít jak UART tak SPI rozhraní. Z hlediska vývoje vestavných (embedded) zařízení v této aplikaci, není rozhraní USB podstatné. Obě z výše uvedených komunikačních portů poskytují možnost, změnit tovární nastavení parametrů v oblasti primární databáze parametrů (Persistent Store Keys). U rozhraní UART je nutné jen upravit příslušné napěťové úrovně pomocí integrovaného obvodu MAX2323, jak již bylo uvedeno na obr.26. U rozhraní SPI je situace podobná, opět je nutné upravit napěťové úrovně, ale v jiném rozsahu. Nutná konverze napěťových úrovní byla přehledně uvedena v tab. 5. Dalším problémem je to, že standardní osobní počítač nedisponuje externím portem pro SPI komunikaci, proto je s výhodou využito portu paralelního, na kterém lze toto rozhraní softwarově emulovat. Pro převod napěťových úrovní je vývojový modul vybaven integrovaným obvodem 74LVX245. Tento integrovaný obvod se využívá jako budič sběrnice s funkcí step-down, tzn. na výstupu tohoto obvodu je vždy maximálně 3.3 *V* což je odvozeno od jeho napájecího napětí. Návrh vývojového modulu byl z části veden podle doporučeného zapojení modulu Bluetooth uvedeného v [4].

Kromě těchto dvou významných rozhraní je možné k vývojovému modulu připojit i mikrokontrolér PIC, který obsahuje rovněž rozhraní UART, ale podle tab. 5 opět přes budič sběrnice 74LVX245 určený pro úpravu rozsahu signálu. Další vlastnost, kterou tento modul poskytuje je možnost odpojení napájecího napětí modulu Bluetooth, pomocí tranzistoru MOSFET (IRLML6302) a vynutit tak obnovení výchozího stavu modulu Bluetooth. Ve standardním stavu je tento tranzistor sepnut a poskytuje napájení pro modul Bluetooth. V případě chybného řízení modulu Bluetooth, dojde k uváznutí tohoto modulu v nedefinovaném stavu a je nutné ho uvést do stavu výchozího, proto je pro vývoj tato funkce nezbytná. Řízení spínacího tranzistoru je vyvedeno na výstupní pin vývojové desky a je možné ho přímo ovládat z mikrokontroléru nebo přiložením napětí 3 – 5 *V*. Napájení celé vývojové desky, včetně modulu Bluetooth, je řešeno pomocí externího zdroje napětí, neboť odběr modulu Bluetooth je až 160 *mA*. Velikost tohoto proudu tedy znemožňuje napájet celý systém z rozhraní osobního počítače. Výjimku tvoří pouze rozhraní USB, které je schopné poskytnout až 0.5 *A*, ale toto rozhraní vývojový modul neposkytuje. Napájecí napětí tohoto

modulu může být v rozsahu 5 – 10 V, potřebné napětí 3.3 V tedy zajistí stabilizátor napětí LF33, který je schopen poskytnout až 1 A. Schéma zapojení vývojového modulu je uvedeno v příloze A.2 a obrazec plošných spojů s popisem jednotlivých vstupů a výstupů je v příloze A.1.

3.3.4 Mikrokontrolér PIC16F873

Jádrem celého řídicího systému je mikrokontrolér PIC 16F873 v provedeni SMD od firmy Microchip. Jde o mikrokontrolér, který je určen pro nenáročné aplikace s jednoduše dostupným vývojovým prostředím poskytovaným výrobcem. Detailní popis je uveden ve specifikaci [10]. Zde budou uvedeny jen významné vlastnosti vztahující se k funkčnosti celého zařízení a implementaci softwaru.

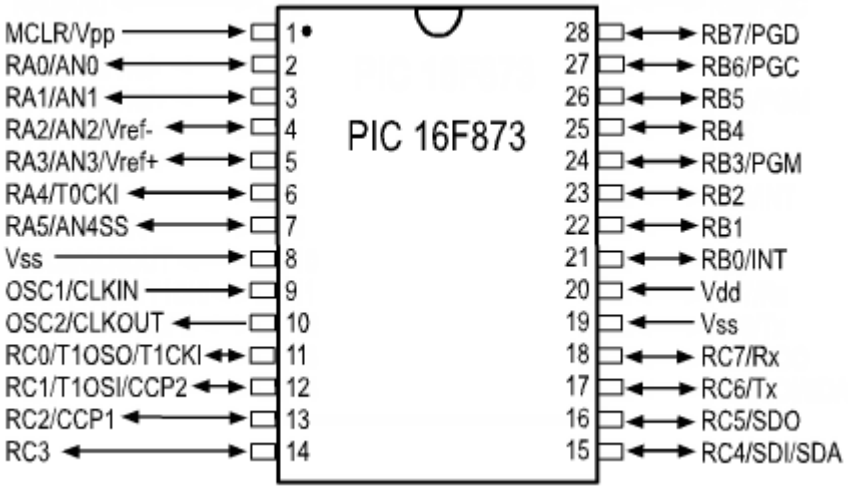
Základem mikrokontroléru PIC 16F873 je 8 - bitový mikroprocesor (CPU) postavený na střadačové architektuře s jádrem RISC (Reduced Instruction Set Computer – redukovaná instrukční sada, neobsahuje složité vícecyklové instrukce). Nominální taktovací frekvence tohoto mikrokontroléru je 4 až 20 *MHz*. Provedení jedné jednocyklové instrukce vyžaduje čtyři hodinové takty mikroprocesoru, tzn. že doba vykonání jedné instrukce je rovna 200 *ns* (pro 20 *MHz*) až 1 μs (pro 4 *MHz*). Centrální procesorová jednotka mikropočítače má na svém čipu umístěny 4 *kB* programové paměti, 192 bajtů paměti datové pro dočasné uložení pracovních parametrů a hodnot. Kromě těchto dvou základních pamětí je mikrokontrolér vybaven také 128 bajty paměti EEPROM (pro účely této práce nebyla využita). Pro komunikaci s okolím využívá tři vstupně/výstupní porty. Port A,B a C, které jsou po resetu mikrokontroléru konfigurovány jako vstupní brány. Mikrokontrolér je vybaven řadou důležitých periferních obvodů. Obsahuje tři jednotky čítač/časovač (dvě osmibitové, jednu šestnáctibitovou), dvě jednotky input capture/output compare, obvody pro sériovou komunikaci USART/SPI a desetibitový AD převodník s pěti multiplexovanými kanály. Na obr.29 Je uveden popis pinů a umístění jednotlivých periférií na vstupně/výstupní piny mikrokontroléru.

Přehledový popis jednotlivých portů:

Port A: pouze šestibitový vstupně (TTL), výstupní (CMOS) port, RA0 – RA3 a RA5 jsou po resetu konfigurovány jako analogové vstupy, lze je však nastavit i jako digitální vstupy/výstupy (nastavení reg. ADCON0 a ADCON1), na vstupu RA4 (pouze číslicový) je umístěn schmittův klopný obvod.

Port B: oproti portu A je až osmibitový (TTL/CMOS), RB0 poskytuje externí přerušení programu, piny RB7, RB6 a RB3 jsou využívány pro programovaní mikrokontroléru.

Port C: opět osmibitový vstupně (TTL)/výstupní (CMOS) port, schmittovy klopné obvody, poskytují také přístup na vnitřní periférie USART, SPI, CCP1, CCP2 a pin RC0 pro externí vstup časovače 1.



Obr. 29 popis jednotlivých pinů mikrokontroléru PIC 16F873.

Organizace paměti

Organizace paměti je uvedena na obr. 31. Jde o rozdělení programové paměti a stavbu zásobníku.

Paměť programu

Jak již bylo výše uvedeno, obsahuje mikrokontrolér 16F873 4 *kB* programové paměti. I když rodina 16F87x je schopna adresovat paměť 8 *kB*, instrukce skoků dovoluje adresovat

maximálně jedenáct bitů. Což omezuje adresování na blok o velikosti 2 *kB*. Adresový prostor je tedy rozdělen na dvě programové stránky (page0, page1). Programový čítač je třináctibitový, ale jen osm bitu (PCL) je přímo adresovatelných. Zbylých pět bitů je přístupných přes nepřímé adresování pomocí reg. PCLATH. Je-li nutné využít instrukce skoků (CALL, GOTO) mezi jednotlivými programovými stránkami, je třeba provést příslušné změny v adresovém reg. PCLATH, který ukazuje na vrchní část (5 bitů) třináctibitového programového čítače.

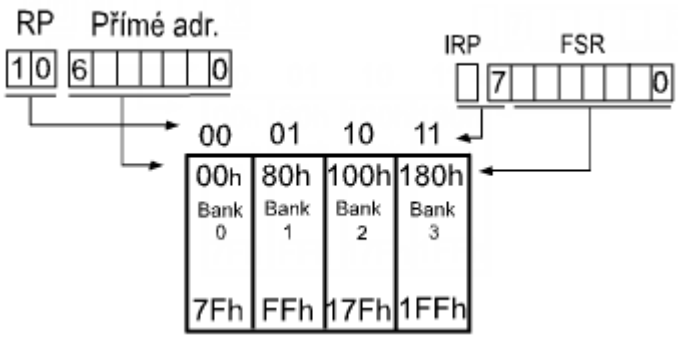
Zásobník

U rodiny mikrokontroléru 16F87x je implementován cyklický zásobník s osmi úrovněmi. Z hlediska čtení či zápisu není tento zásobník přístupný pro programátora. Je určen pouze pro ukládání návratových adres při volání podprogramu pomocí instrukcí CALL a při skoku do obslužné rutiny přerušení. Počet těchto vnoření je tedy omezen na osm, při devátém vnoření dojde k přepisu nejstarší adresy uložené na zásobníku a program ztratí možnost se správně navrátit z těchto podprogramů. U tohoto procesoru je tedy vyloučena rekurze a také předávání parametrů funkcím pomocí zásobníku.

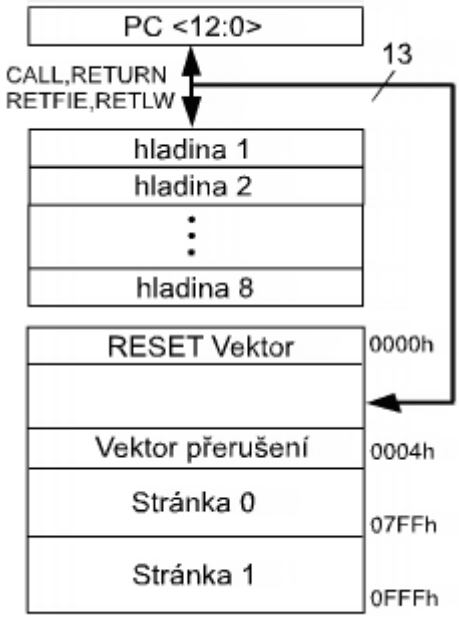
Stránkování datové paměti

Datová oblast paměti tohoto mikrokontroléru je rozdělena na dvě banky (bank0, bank1) Další dvě banky (bank2, bank3) nejsou přístupné pro zápis a jejich hodnoty jsou tvořeny zrcadlením bank 0 a 1. Instrukce pro operace s datovými registry adresují paměť pouze sedmi bity, proto je nezbytné pro přístup na adresy od 7Fh přepnout datovou banku v reg. STATUS (bity RP0 a RP1).

Kromě tohoto přímého adresování (obr. 30) podporuje mikrokontrolér i adresování nepřímé, pomocí adresového reg. FSR a virtuálního reg. INDF. Adresový reg. FSR je v tomto případě osmibitový a poskytuje přístup až na adresu FFh, což odpovídá paměťovému rozsahu dvou datových bank. Pro přístup do datové banky 2 a 3, tedy nad adresy dostupné FSR reg., je nutné nastavit bit IRP v reg. STATUS. Hodnoty na adrese určené FSR reg. a bitem IRP je možné přečíst z virtuálního reg. INDF mapovaného na adresovanou buňku



Obr. 30 přímé a nepřímé adresování v PIC 16F873.



Obr. 31 organizace programové paměti a zásobníku.

4. Kapitola

Návrh a implementace komunikačního softwaru

V této části diplomové práce bude uveden návrh a popis komunikačního softwaru pro obě strany komunikace. Ovládací a řídicí program osobního počítače a program do mikrokontroléru zajišťující komunikaci s tímto osobním počítačem přes bezdrátové spojení pomocí modulů SieMoS50037. Poslední část kapitoly se zabývá návrhem programu pro komunikaci mezi dvěma osobními počítači.

4.1 Popis komunikačního protokolu modelu

Komunikace s modelem probíhá v režimu maser – slave. Řídicí počítač jako master pomocí příkazů ovládá jednotlivé pohony nebo řídí odměry některých veličin (vzdálenost, náklon, azimut atd.). Komunikace probíhá pomocí výměny paketů jejichž formát je uveden v [3]. Každá odpověď je uvozena dvěma synchronizačními znaky 0xFE a 0xFF a dvěma 0xFF a 0xFF znaky je také ukončena. Podle těchto znaků je detekováno přijetí kompletního paketu. Na obr.32 je standardní rámec příkazu a na obr. 33 je standardní rámec odpovědi.

0xXX	0xLL	data

Obr. 32 standardní formát příkazu pro řízení modelu.

První bajt (*XX*) je adresa zařízení, následovaná bajtem, který určuje délku dat. Ta jsou umístěna v poli *data*.

0xFE	0xFF	0xXX	data	0xFF	0xFF

Obr. 33 standardní typ odpovědi na příkaz pro řízení modelu.

První a druhý bajt tvoří synchronizační bajty, ty jsou následovány adresou zařízení. Dále následují jen data, která jsou ukončena opět synchronizačními znaky (toto uskupení znaků je v komunikaci jedinečné).

4.2 Program pro mikrokontrolér PIC 16F873

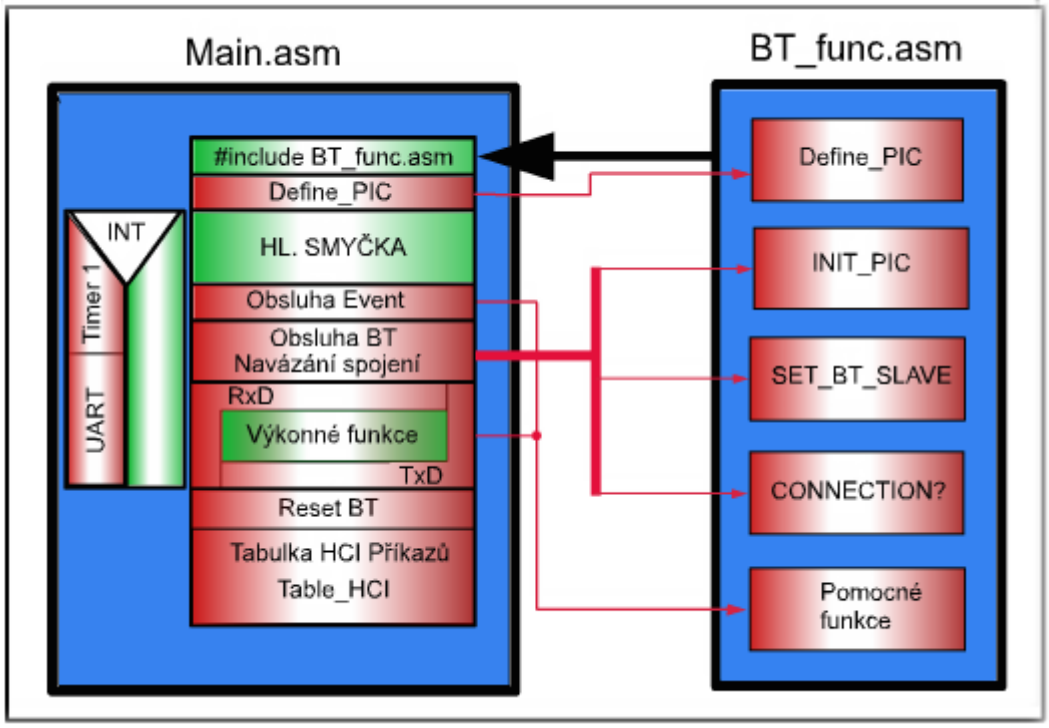
Zapojení a vnitřní paměťová struktura mikrokontroléru PIC 16F873 byla popsána v předchozí kapitole. Zde bude uveden vnitřní ovládací software, ale jen část týkající se komunikace pomocí bezdrátové technologie Bluetooth s využitím modulů SieMo od firmy Siemens. Druhá část programu zabývající se obsluhou všech periférií modelu bude uvedena jen z nezbytně nutné části. Detailní popis ovládacího programu pro periférie modelu je uvedena v [3].

Pro programování a výsledné ladění byl využit originální programátor MPLAB – ICD dodávaný výrobcem. Všechny funkce a makra byly napsány a odladěny ve vývojovém prostředí MPLAB 5.7 spol. Microchip. Jde o volně dostupný software a společně s ostatním je uložen na přiloženém CD ve verzích 5.7 a 6.3.

4.2.1 Struktura programu

Jak již bylo uvedeno výše, je jádrem celého řídicího systému v modelu vzducholodi mikrokontrolér PIC 16F873. S ohledem na velikost programové paměti, rychlost vykonávání instrukcí a optimálnost zdrojového kódu, byl řídicí program napsán v absolutním assembleru. Absolutním proto, že zdrojová část kódu neobsahuje žádné další přilinkované moduly v předkompilované formě a tedy žádné části kódu nejsou umístěné na relativních adresách.

Návrh struktury programu pro bezdrátové spojení byl podřízen požadavku zajišťující snadnou integraci a přenositelnost do nového zdrojového kódu. Proto byla část programu pro obsluhu komunikace s modulem SieMo, který zajistí bezdrátové spojení, navržena jako přídavný programový modul (opět na úrovni absolutního assembleru). Návrh tohoto programového modulu není záležitost zcela separátní, ale je velice úzce spjata s návrhem hlavního programu pro obsluhu a řízení periférií. Provázanost celého systému je znatelná zejména v části pro zpracování obsluhy přerušení. Mezi další části úzké spolupráce je rozdělení datové paměti. Je třeba přesně určit kam se budou ukládat aktivní data a kde budou umístěny pomocné proměnné, řídicí registry a příslušné příznaky např. pro zjištění stavu komunikace. Integrace modulu pro obsluhu bezdrátové komunikace a jeho součástí je pro snazší orientaci uvedena na obr.34. Je zde uvedena nezbytná provázanost modulu, který využívá stejných součástí jako hlavní program (především jde o sdílení periférií).



Obr. 34 struktura programu pro PIC 16F873.

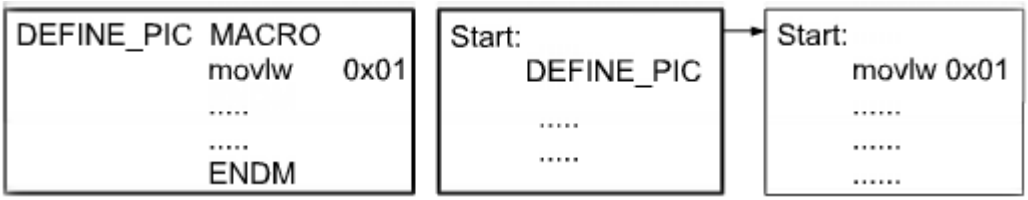
Na obr.34 je zeleně označena část hlavního programu pro obsluhu periférií modelu. Červeně je vyznačena část spadající do oblasti obsluhy komunikace přes bezdrátové rozhraní. Z tohoto obrázku je rovněž patrné, že se obě části dělí o společný vektor přerušení. Jak od časovače 1 tak od sériového kanálu. Všechny funkce poskytované knihovnou *BT_func.asm* využívají ke své funkci blok proměnných, které jsou umístěné v bance 1 (obr.30) datové paměti. Všechny ostatní proměnné, které jsou využity pro chod hlavního programu jsou umístěny v datové oblasti banky 0.

I když náplní této kapitoly není detailní popis proměnných využívaných hlavním programem či knihovnou *BT_func.asm* (všechny definice jsou uvedeny na začátku programu) je vhodné upozornit alespoň na proměnnou zajišťující přístup obou částí programu k systémovým prostředkům, které poskytuje mikrokontrolér PIC (timer, UART). Tento řídicí registr je umístěn v datové paměti banky 0 pod názvem *BT_reg*. Registr obsahuje tři významné bity. Bit pojmenovaný *Semf* (BTreg <0>) určuje, zda jsou systémové prostředky využívány a obsluhovány hlavním programem (bit nulový) či využívány knihovnou (bit jedničkový). Druhý významný bit *DReady* (BT_reg <1>), který signalizuje příchod a tedy následné zpracování příchozí události od zařízení Bluetooth, která je vygenerována jako

reakce na aktuálně odeslaná data. Třetí bit Online (BTreg <2>) pouze označuje zda úspěšně proběhlo spojení se vzdáleným masterem a zda je komunikace připravena.

4.2.2 Popis knihovny BT_func.asm

Programový modul pro komunikaci s modulem SieMo je složen ze čtyř důležitých částí, které jsou využívány hlavním programem. V knihovně jsou tyto jednotlivé bloky implementovány jako makra. Makrem je zde myšlena část programu, určená pro jednorázové volání. Makro (část programového kódu) je zkopírováno na všechna příslušná místa v programové paměti, která jsou označena jeho názvem (ukázka obr.35). Použití těchto maker bylo užito většinou za účelem uvedení mikrokontroléru do definovaného nastavení, provádějící se pouze na začátku komunikace s Bluetooth modulem a poté pro navázání komunikace se vzdáleným zařízením. Pro vícenásobné volání je, z hlediska úspory programové paměti, lépe využít volaní podprogramů instrukci CALL. Tyto makra a potřebné funkce jsou definovány v souboru *BT_Func.asm*, jež se pomocí direktivy přilinkují k hlavnímu programu pro obsluhu periférií.



Obr. 35 definice a vložení makra do programu.

Přilinkování knihovny BT_func.asm

Je-li nutné využívat funkce a makra umístěné v knihově (jakékoliv), je nutné tuto knihovnu k hlavnímu programu nějakým způsobem přiřadit. Toto se provede programovým příkazem #include "BT_func.asm ". Význam tohoto řádku spočívá v tom, že linker od adresy, na které je umístěn tento příkaz umístí celý obsah knihovny. Je tedy nezbytně nutné umístit tento příkaz až za návěští ORG 004, kde je umístěn vektor přerušení. Jinak dojde k přepsaní tohoto vektoru některou instrukcí z knihovny, popřípadě i počáteční části hlavního programu. Některé překladače na tuto chybu upozorní některé nikoliv.

Popis součástí knihovny

Zde bude uvedena přehledová charakteristika významných maker a funkcí, které knihovna obsahuje. Nejvýznamnější jsou makra uvedena na obr.34.

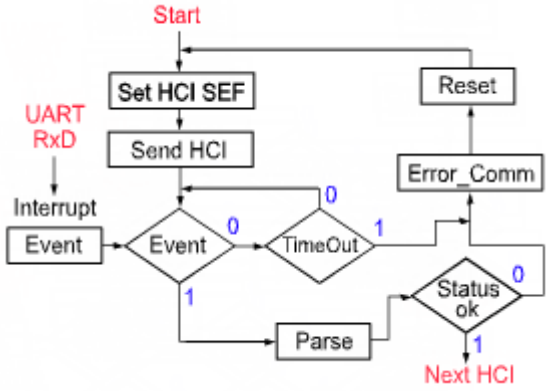
Define_PIC: první nezbytná část programu, obsahuje všechny definice proměnných, které jsou využívány knihovnou. Dále pak inicializaci všech konstant (adresy bufferů, délky příkazů, adresy příkazů). Umístění tohoto makra v hlavním programu je do jisté míry libovolné. Pro ucelenost a přehlednost bylo vloženo na konec vlastních definic proměnných a konstant hlavního programu.

INIT_BT: Toto makro zajišťuje potřebné nastavení I/O portů, periférií pro sériovou komunikaci, nastavení přerušovacího systému a konfiguraci časovače 1. Dále pak inicializaci všech definovaných proměnných a nulování paměťového bloku pro příjem dat přes UART. Poslední část tohoto makra povolí napájení Bluetooth modulu, nastavením logické úrovně 0 na pinu RA4. Následně pak zajistí bezpečný start s časovým zpožděním pro korektní inicializaci, kterou Bluetooth modul vyžaduje.

SET_BT_SLAVE: hlavní náplní tohoto makra je nastavení Bluetooth modulu SieMo do stavu *Inquiry scan*, aby bylo možné modul vyhledat masterem a spojit se sním. K tomu jsou využity HCI příkazy, které byly popsány v [1]. Definice těchto příkazů jsou umístěny v tabulce na konci hlavního programového kódu, jak uvádí obr. 34. (Table_HCI). Při vykonávání instrukcí tohoto makra jsou periférie a přerušovací systém přiděleny knihově *BT_func.asm* (BTreg <Semf> = 1), protože pro odesílání a následnou kontrolu HCI příkazů je využíváno sériového rozhraní mikrokontroléru. K nastavení modulu Bluetooth je třeba čtyřech HCI příkazů (set event filter, write page scan activity, write inguiry scan activity, write scan enabled). Pro názornost je na obr. 36 uveden vývojový diagram tohoto makra, ale pouze pro jeden příkaz, protože ostatní tři odpovídají stejné politice zpracování a vyhodnocení.

První částí diagramu tvoří blok Set HCI SEF zajišťující přípravu příslušného HCI příkazu a jeho následné odeslání přes sériové rozhraní mikrokontroléru. V tomto případě jde o příkaz Set Evnet Filter. Následně se čeká na potvrzení správnosti odeslaného paketu (Command Complete event) od modulu Bluetooth (Host Controller). Čas přijetí je limitován

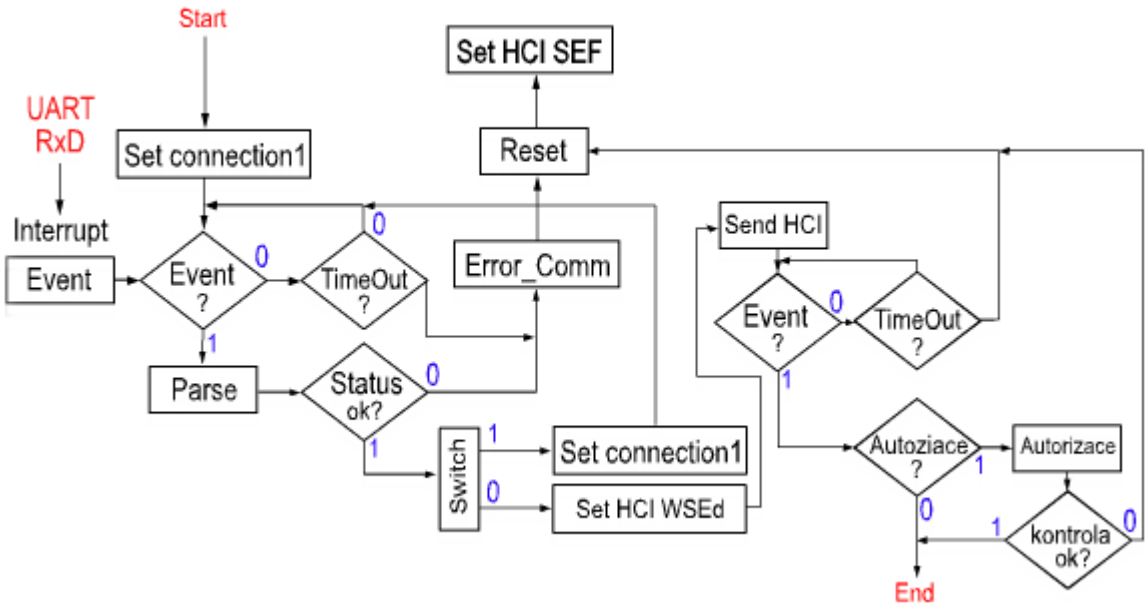
určitou hranicí po jejímž překročení se předpokládá chybné doručení příkazu a nastává restart modulu. Tato situace je generována i při přijetí zprávy označující chybu (chybový status bajt). Chybový kód je uložen v registru Error_Comm. Je-li příkaz úspěšně zpracován a vykonán pak přijatá zpráva, může kromě status bajtu obsahovat také parametry pro nastavení modulu Bluetooth (zajišťuje blok pars). Jsou-li tyto parametry ve zprávě umístěny, jsou vyjmuty a bezpečně uloženy do datové paměti mikrokontroléru.



Obr. 36 vývojový diagram makra SET_BT_SLAVE.

CONNECTION?: poslední makro, které knihovna obsahuje. Zajišťuje navázání spojení s masterem a to tím způsobem, že ve smyčce čeká na speciální paket, který vyšle modul Bluetooth (Host Controller) mikrokontroleru (Host). Ten je následně rozdělen na části a vyhodnocen. Je-li vše úspěšné, je spojení navázáno. Poslední úkol tohoto makra je vypnout scanovací režim modulu SieMo, aby se nehlásil jako slave pro jiný master. To se provede opět příslušným HCI příkazem [1]. Po úspěšném navázání spojení je možné v rámci tohoto makra odeslat přes bezdrátové spojení datový paket a vyčkat na specifickou odpověď od mastera. Tato akce je v programu uvedena pro kontrolu navázaného spojení a lze ji také s výhodou využít jako autorizaci správného mastera. Tuto autorizaci je třeba zapnout před využitím makra *CONNECTION?* a to nastavením bitu *autoriz* registru *BT_Status* (BT_Status <2>). Dojde-li k úspěšnému spojení je od zařízení slave poslán požadavek na autorizaci ve formě datového paketu obsahujícího data 01h, 02h, 03h, 04h. Pro úspěšnou autorizaci očekává slave od mastera opět datový paket v jehož datové části je uložena sekvence 05h, 06h, 07h, 08h. Toto složení autorizačních bajtů je pouze zkušební a lze jednoduše změnit. Změna požadavku na autorizaci lze provést v Table_HCI. Příslušná odpověď je pak uvedena

ve funkci *Check_Code* umístěné v *BT_func.asm*. V případě chybné autorizace dochází k restartu modulu Bluetooth a je nutné se vzdáleným modulem opět navázat spojení. Vývojový diagram makra *CONNECTION?* je uveden na obr. 37.



Obr. 37 vývojový diagram makra *CONNECTION?*.

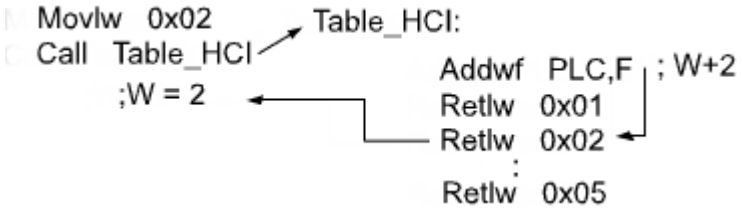
Pomocné funkce: tato část zahrnuje všechny ostatní funkce, které soubor *BT_func.asm* obsahuje. Kromě předchozích maker je v tomto bloku začleněna i obslužná rutina pro příjem a odesílaní dat přes sériové rozhraní, funkce pro doplnění datových rámců o příslušné úvodní hlavičky, které vyžaduje Bluetooth a kontrolu událostí na odeslané datové rámce. Paří sem i méně významné funkce např. pro nulování, přednastavení registrů, či kontrolu autorizačních bajtů.

Všechna makra, funkce a jejich vývojové diagramy jsou uvedeny pouze v přehledové formě a tedy bez přesné implementace v mikrokontroléru. Detailní popis není třeba zde uvádět, protože program je dostatečně komentován. Hlavní program *Main.asm* a knihovna *BT_func.asm* je umístěna na přiloženém CD. Mezi součásti umístěné v hlavním programu patří blok reset a tabulka HCI příkazů.

Blok reset náležící do hlavního programu pouze volá funkci reset, která na krátký časový interval odepne modul od napájení a po návratu z této funkce provede skok na návěští, kde je umístěno makro INIT_BT. Tím má modul opět možnost se spojit s masterem.

Tabulka HCI příkazů

Poslední část, umístěná v hlavním programu je tabulka HCI příkazů. Jde o funkci, která jako vstupní parametr, uložený v pracovním registru W, nese informaci o relativním posunu programové čítače. Uvnitř této funkce se pak provede součet aktuální hodnoty programového čítače a pracovního registru. Po opětovném uložení do programového čítače se provede skok v bloku instrukcí RETLW pro návrat, ovšem s uloženou hledanou hodnotou v reg. W z této tabulky. Funkce tabulky je ukázána na obr. 38. Výhoda toho postupu spočívá především v úspoře instrukcí, optimalizaci zdrojového kódu, přehlednosti a využití pouze programové paměti. Nevýhoda spočívá v tom, že tabulka musí být uložena na přesně stanovené adrese v programu a její délka může být maximálně 256 hodnot.



Obr. 38 způsob implementace tabulky HCI příkazů.

4.3 Hlavní řídicí program

Tato část práce se zabývá návrhem hlavního řídicího programu pro model vzducholodi. Program je umístěn v osobním počítači a pomocí bezdrátového spojení komunikuje s modelem. Modul je k osobnímu počítači připojen přes sériové rozhraní podle obr.26. Řízení a komunikace s modulem *SieMo* probíhá opět na úrovni vrstvy HCI. Je tedy využíváno HCI příkazů, které jsou implementovány v hlavním programu. Řídicí program byl napsán ve vývojovém prostředí Borland C++ Bulider ver. 5.0 Enterprise.

4.3.1 Přístup k sériovému portu

Pro nezávislost na platformě operačního systému WINDOWS bylo zvoleno řešení přístupu na hardwarový blok pomocí aplikačního rozhraní daného operačního systému. Tím získáme z pohledu programátora nezávislost na hardwarovém řešení dané periférie.

Operační systém je striktně multitaskingový a nabízí tak možnost zpracování jednotlivých procesů paralelně. Ve WIN32 je na sériový port po jeho otevření hleděno jako

na soubor, který je otevřen pro čtení i pro zápis. Platforma WINNT využívá pro obsluhu periférie chráněného módu procesoru a tak není možné přímý přístup na hardware periferie. Je tedy nutné přistoupit na periferii přes ovladač zařízení. Obsluhu sériového potru je možné rozdělit do několika částí:

1. Vytvoření objektu obsluhy sériového portu.
2. Otevření portu a získáni handle pro přístup.
3. Nastavení pracovních parametrů portu.
4. Obsluha čtení a zápisu.
5. Obsluha jednotlivých událostí.

Pro sériovou komunikaci pod WIN32 byla využita knihovna funkcí pro obsluhu sériového portu, která byla naprogramována podle [11]. V této knize je i výčet všech potřebných API funkcí a příslušných struktur.

4.3.2 Struktura řídicího programu modelu

Struktura programu obsahuje čtyři hlavní části. Část pro řízení modelu, část pro vizualizaci naměřených dat, část pro komunikaci a část přístupu k sériovému portu. Kromě části řízení a vizualizace, která je tvořena hlavním vláknem aplikace, má každá část své vlákno. Jednotlivé části si předávají data pomocí struktur a zpracování těchto dat je řízeno pomocí bitových proměnných.

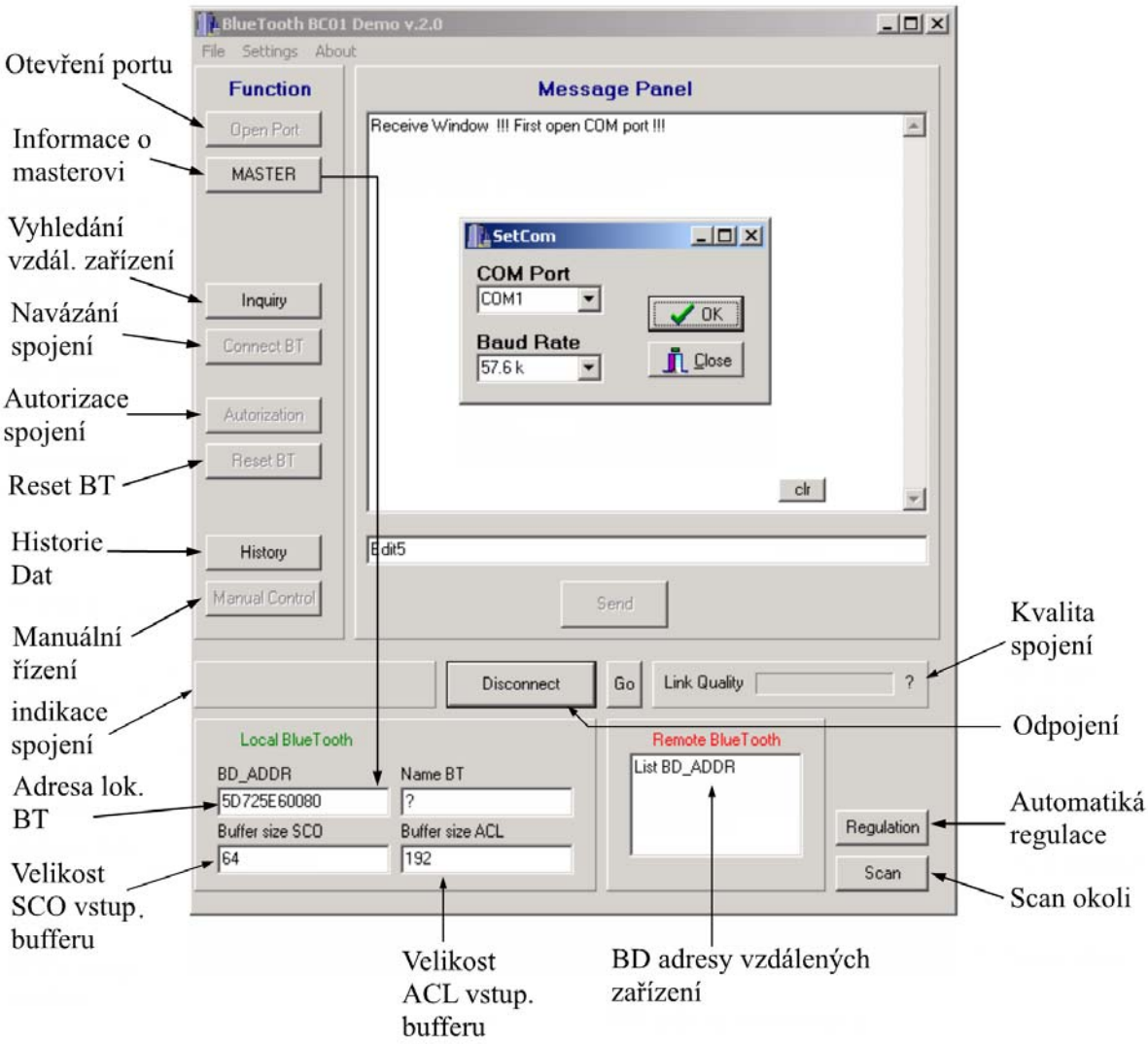
Celá struktura programu je na obr. 40. Jsou na ní vyznačeny nejvýznamnější bloky programu. Oblast vizualizace a řízení týkající se komunikace přes Bluetooth, je uvedena v hlavním formuláři (*BTmaster.cpp*), který se zobrazí po nastavení a otevření sériového portu. Ve stejné programové jednotce je umístěn přijímací buffer (pole typu unsigend char), do kterého jsou zapisována přijatá data, určena ke zpracování. Následuje blok (ParsBlok) pro identifikaci a rozdělení paketů. Takto získaná data jsou určeny pouze pro řízení komunikace prostřednictvím modulu *SieMo* (Bluetooth) a důležité parametry (BD_addr, ClockOffset atd.) jsou uchovávány v datové struktuře *BT_param*, která je součástí objektu *HCI_Command*. Tento objekt rovněž definuje metody pro řízení komunikace pomocí HCI

příkazů. Tento objekt je velice úzce propojen s objektem *Events*, který je určen pro zpracování a vyhodnocení událostí příslušející k danému příkazu. Tyto dvě části spolu s programovou jednotkou *Comm.cpp* zajišťují komunikaci s modulem SieMo. Je-li vytvořeno datové spojení s modelem a je-li kladně potvrzena autorizace přístupu k modelu, je možné model ovládat i řídit.

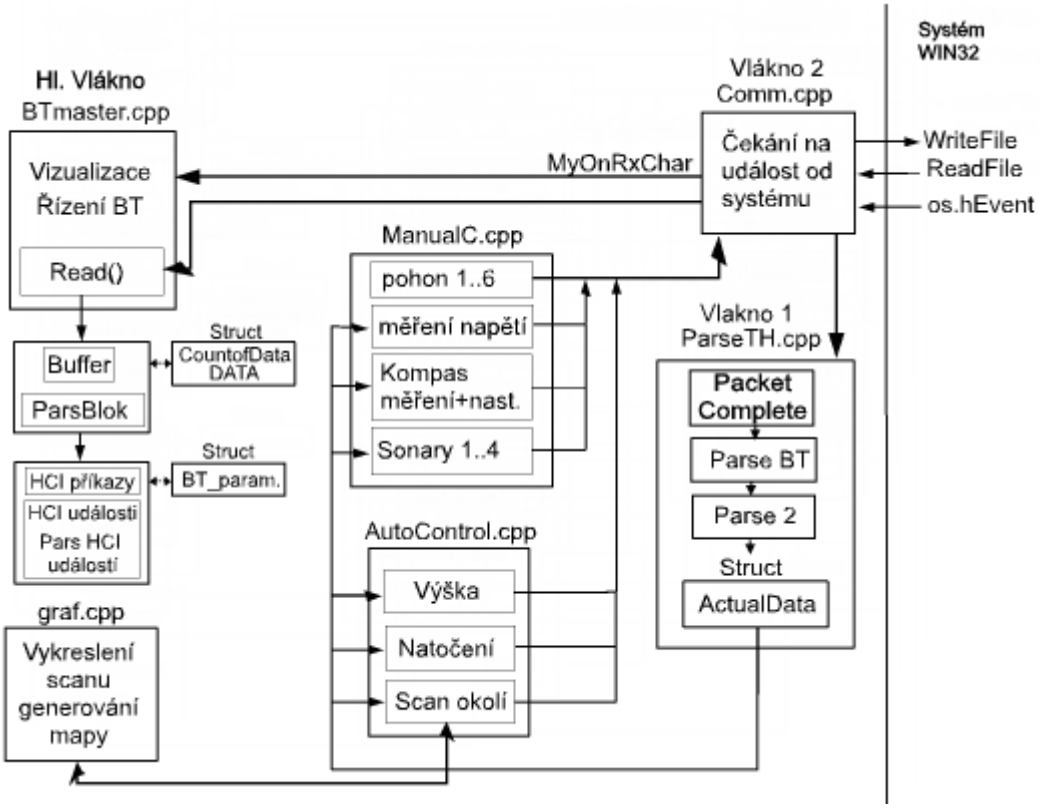
Při ovládání a řízení modelu je využívána programová jednotka *ParsTh.cpp*. Hlavní část této programové jednotky tvoří vlákno pro ukládání přijatých datových paketů do vlastního datového pole (ByteBuffer). Toto pole je součástí struktury, ve které jsou umístěny ukazatele na první nepřečtený bajt a na poslední aktuálně uložený bajt. Přístup do této datové struktury je sdílen více vlákny, proto je k synchronizaci přístupu do struktury využito kritických sekcí. V režii tohoto vlákna je i identifikace přijatých paketů a jejich rozdělení na části patřící modulu Bluetooth a na část nadřízeného komunikačního protokolu pro ovládání modelu. Významná součást toho programového modulu je funkce *PacketComplete*, která povolí rozdělení paketu až v případě, že je přijata celá datová část paketu. Po oddělení datové složky paketu, jsou příslušná data (bajtová reprezentace) přepočítána na číselné hodnoty a uložena do struktury *ActualData*. Tím je ukončen příjem dat a je nastavena příslušná bitová proměnná odpovídající datové složce, která byla aktualizována.

Ovládání modelu zajišťuje programová jednotka *ManualC.cpp* patřící do hlavního vlákna aplikace. V této jednotce jsou sestaveny příslušné příkazy přímé ovládání periférií modelu. V této části není implementována zpětná vazba.

Úvodní vzhled aplikace je uveden na obr. 39 Formulář na obr. 39 zajišťuje správu komunikace, nalezení a připojení ke vzdálenému zařízení, autorizaci spojení a možnost další volby z nabízených funkcí. Jednotlivé formuláře jsou uvedeny v dodatcích C.1 a C.2.



Obr. 39 hlavní formulář aplikace.



Obr. 40 struktura hlavního řídicího programu.

Pro automatické řízení modelu je naprogramována jednotka *AutoControl.cpp,* opět náležící hlavnímu vláknu. Tato jednotka obsahuje stavový automat pro zpětnovazební řízení modelu. Stavový model řízení spočívá v odeslání dotazu na některou z veličin jejíž hodnota je regulována a následně očekává naměřená data. Příjem dat není realizován pomocí čekací funkce, ale je konstruován s využitím ukazatelů na funkci. Po přijetí a identifikace očekávaných dat jsou tato data umístěna do struktury *ActualData* a podle nastavení bitové proměnné se určí, zda jde skutečně o nová data nebo zda je to jen potvrzení akčního zásahu. V případě, že přijatá data jsou skutečné naměřené hodnoty, je pomocí regulátoru PID (implementován jako objekt), vypočten akční zásah, který se jako parametr uloží do datové části paketu ovládající pohony modelu a následně je odeslán přes bezdrátové rozhraní do modelu. V této chvíli se očekává příjem potvrzení, že akční zásah byl proveden. Je-li toto potvrzení přijato je odeslán požadavek na změření další aktuální hodnoty (kompas, ultrazvukový dálkoměr, napětí atd.).

Mezi jednotlivými příkazy je stanovena pevná čekací doba, realizována komponentou *Timer*, která vytváří minimální časové intervaly mezi jednotlivými zprávami. Toto zpoždění

je nutné pro udržení stabilní komunikace, protože obě strany komunikace nemají stejnou dobu zpracování instrukcí. V případě, že by mezi zprávami nebyl umístěn tento zpožďovací interval, nastane situace, že program mikrokontroléru není navrácen ze všech volaných podprogramů a nejsou inicializované příslušné proměnné. Následným voláním přerušení dojde k přetečení zásobníku.

Poslední významnou částí je programový modul pro zpracování a vizualizaci obrazu okolního prostředí, které bylo sejmuto pomocí ultrazvukových dálkoměrů, umístěných vespod modelu. Část zpracování a vykreslení jednotlivých odrazů je uvedena v [3].

4.4 Program pro komunikaci mezi dvěma počítači

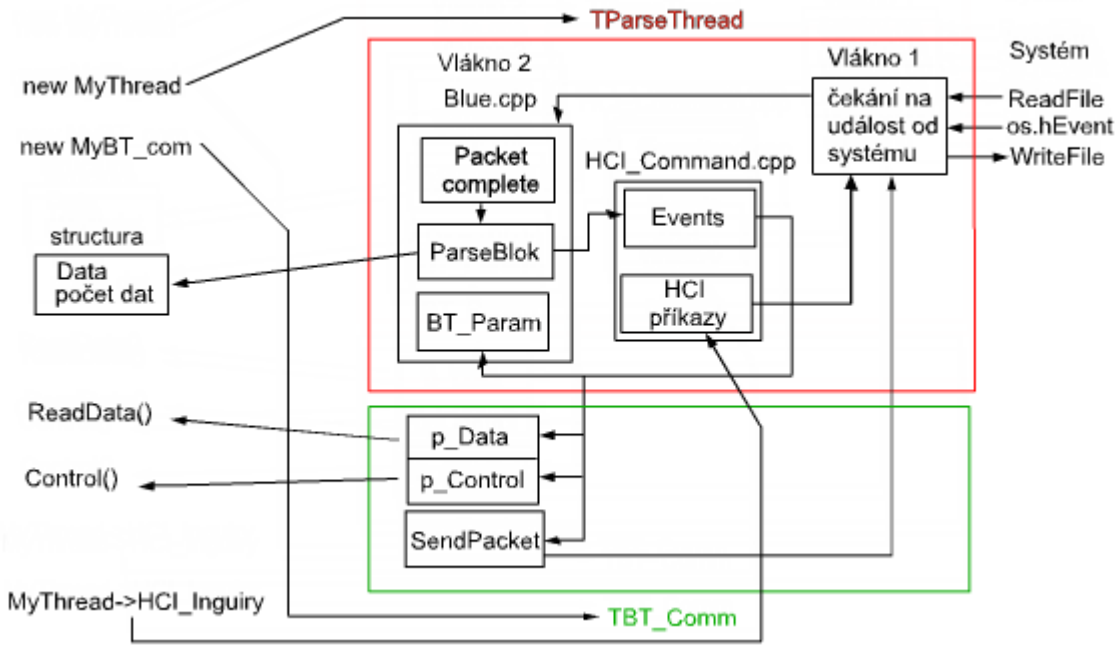
Tato část se věnuje popisu programu pro komunikaci mezi dvěma osobními počítači s využitím protokolu Bluetooth a modulů SieMo. Návrh toho programu byl podřízen požadavku na oddělní protokolových vrstev a vytvořit tak programovou jednotku, která by programátorovi umožnila využití modulu Bluetooth v jiné aplikaci. Tato jednotka byla naprogramována pro spojení dvou osobních počítačů.

4.4.1 Popis struktury programu

Hlavní část programu zabývající se řízením komunikace s využitím modulů Bluetooth je umístěna v programové jednotce *Blue.cpp*. Ta zajišťuje veškerou komunikaci a řízení modulu Bluetooth s využitím HCI příkazů. Struktura programu je uvedena na obr. 41. Tato jednotka využívá další programovou jednotku, ve které jsou uvedeny metody pro zpracování událostí a příkazy HCI vrstvy. V jednotce *Blue.cpp* je definice struktury, ve které jsou uloženy všechny důležité informace o lokálním a vzdáleném zařízení. Jednotka *Blue.cpp* obsahuje třídu *TParseThread*, která tvoří jádro zpracování dat a událostí. Dále jednotka *Blue.cpp* obsahuje třídu TBT_Comm, která zajišťuje odesílání a přijímaní datových paketů. Vytvoření dvou objektů těchto tříd zajistí zpracování paketů na úrovni HCI ovladače. Pro řízení je opět využito HCI příkazů, které jsou definovány v programové jednotce *HCI_Command.cpp* spolu s třídou pro HCI příkazy je v této jednotce umístěna třída pro obsluhu událostí. Parametry těchto událostí jsou ukládány do datové struktury *BT_Param* která je založena v jednotce *Blue.cpp*. Zpracování událostí tedy probíhá na úrovni vlákna umístěného v jednotce *Blue.cpp*. Při zpracování datových paketů je využito datové struktury,

která je založena v hlavním programu a její typ (StructDataPacket) je definován v hlavičkovém souboru *Blue.h*. Příjem dat pak zajišťuje objekt, který je třídy *TBT_Comm*. Při založení tohoto objektu byly do proměnných uloženy ukazatele na funkce pro zpracování dat, či pro vyhodnocení HCI příkazů, např. spojení navázáno, nejde tedy o HCI události definované protokolem. Vyslání HCI příkazu také není potvrzováno pomocí událostí, ale sama funkce vrátí, zda bylo provedení úspěšné či nikoliv. Stejně tak probíhá odesílání dat pomocí funkce *SendPacket*. Jako parametr této funkce jsou pouze data a jejich počet. Tato funkce sama doplní hlavičku ACL datového spojení a odešle přes sériové spojení k modulu SieMo, který zajistí bezdrátový přenos tohoto paketu.

Na obr. 41 je znázorněno vytvoření objektů datové struktury a obslužných funkcí pro příjem dat. Ukazatele na tyto funkce jsou umístěny v proměnné objektu MyBT_com (podle obr. 41). Ve spodu obr. 41 je uvedeno užití funkce *Inquiry*.



Obr. 41 struktura programu pro komunikaci mezi počítači.

5. Kapitola

Návrh řízení modelu vzducholodi

Tato kapitola popisuje zvolený přístup při identifikaci systému a návrhu jeho řízení. Odhad parametrů systému (časových konstant, dynamického chování) byl proveden pomocí přechodových charakteristik systému. Měření přechodových charakteristik bylo pouze orientační s cílem zjistit základní dynamické parametry systému pro pohyb modelu ve význačných směrech. Pro korektní návrh regulátoru by bylo nutné tuto část realizovat mnohem pečlivěji a doplnit teoretickým rozborem problému s výsledky návrhem modelu a jeho verifikací s reálnou soustavou. Toto ovšem nebylo cílem diplomové práce a vyžádá si značný časový prostor.

5.1 Popis systému

Z hlediska obecné teorie systémů, lze tento model považovat za *otevřený systém*, neboť jeho parametry nezávisí jen na vlastnostech systému, ale také na okolním prostředím. Jelikož převážná část modelu je tvořena zásobníkem plynu (lehčího než vzduch), je chování modelu ovlivněno atmosférickým tlakem a teplotou prostředí. Teplota okolního prostředí není v celém operačním prostoru homogenní a může se v různých částech lišit. Dále na model působí mimo chyb měření také řada dalších poruchových veličin jako např. větrné víry, pokles napětí akumulátoru, únik nosné náplně (hélia).

Podle výše uvedených poznatků je tento model nestacionární, neboť jeho parametry jsou závislé na čase. Aby bylo možné minimalizovat všechny poruchové veličiny, je nutné model umístit do uzavřeného prostoru s konstantní teplotou. Další důležitou částí, kterou lze potlačit nežádoucí vlastnosti, je přesné vyvážení modelu (model samovolně neklesá) s dostatečně výkonným pohonem. Budeme-li předpokládat splnění těchto požadavků pak pro krátký časový úsek lze považovat změnu parametrů modelu za zanedbatelnou. Pak můžeme model považovat za stacionární (t-invariantní).

Model systému je tvořen třemi vstupy a třemi výstupy. Jde tedy o systém MIMO (Multi Input Multi Output [18]). První vstup je tvořen stejnosměrným napětím (PWM modulace) na

hlavních pohonných motorech (vektor rychlosti ve vertikální rovině pohybu). Druhý vstup zajišťuje polohu natočení pohonných motorů podle obr. 22 (rozklad vektoru rychlosti na vertikální a horizontální – podélný pohyb). Poslední vstup je tvořen napětím na směrové vrtuli, která zajišťuje otáčení modelu v horizontální rovině. Hlavní výstupy modelu jsou také tři a to výška modelu (vertikální poloha), natočení modelu vzhledem k severu v horizontální rovině a požadovaná poloha modelu vůči hranicím operačního prostoru (horizontální poloha). Hlavním požadavkem je tedy řízení pohybu v horizontální rovině po stanovené dráze v určité výšce. Předpokládáme-li přesné vyvážení modelu pak regulace výšky není prioritní. Při reálném chování však na model působí poruchové veličiny, které tuto rovnováhu porušují a proto je nutné vždy model doplnit o stabilizaci výšky pomocí regulátoru.

5.1.1 Měření přechodových charakteristik a získání přenosu

Jak již bylo uvedeno v popisu systému, je nutné získat základní informaci o dynamickém chování modelu ve třech základních směrech pohybu: ve vertikálním směru, podélném horizontálním směru a horizontálním natočení modelu. K tomu bylo užito přechodových charakteristik, kde vstupem do soustavy byl vždy jednotkový skok reprezentovaný 70% plněním PWM.

Poloha ve vertikální směru

Je-li obecné těleso umístěno volně v prostoru pak vykovává pohyb kolmý k zemskému povrchu (je přitahováno gravitací země) a jeho aktuální poloha je obecně určena pomocí (1)

$$S = \frac{1}{2} g t^2, \tag{1}$$

kde g je tíhové zrychlení a t čas.

Vztah (1) tedy obecně určuje kvadratickou závislost polohy tělesa na čase. Navíc v tomto případě není zanedbatelný odpor vzduchu a tak vzorec (1) nepopisuje chování modelu ve směru tíhového zrychlení korektně. Nevykonává-li model translační pohyb v horizontální rovině, ale pouze ve směru tíhového zrychlení, je pohybu modelu kladen značný odpor vzduchu, protože tvar modelu není pro tento směr aerodynamický.

Identifikace byla provedena pomocí přechodové charakteristiky, která je uvedena na obr.42. Z tohoto obrázku je patrné, že jde o astatický systém, řád systému byl odhadnut pomocí aproximačního přenosu (2). Určení aproximačního přenosu bylo stanoveno pomocí dynamických parametrů, odečtených z přechodové charakteristiky, které byly následně doplněny do vztahu (2). Aproximační přenos n-tého řádu pro astatický systém je dán:

$$F(s) = \frac{Kv}{p(1+Tp)^n}, \tag{2}$$

kde *Kv* je směrnice tečny asymptoty k přechodové charakteristice, která je pro tento účel normována. Stanovení směrnice *Kv* uvádí vztah (3). *T* je n-násobná časová konstanta určující dynamické parametry modelu. Stanovení této časové konstanty uvádí vztah (4). Řád astatismu lze odhadnout pomocí vztahu (5), kde poměr $h(t_0)/Kv$ je stanoven podle tab.7, která určuje umístění inflexního bodu přechodové charakteristiky. Bod $h(t_0)$ je hodnota přechodové charakteriskty v bodě, kde protíná tečna k přechodové charakteristice časovou osu.

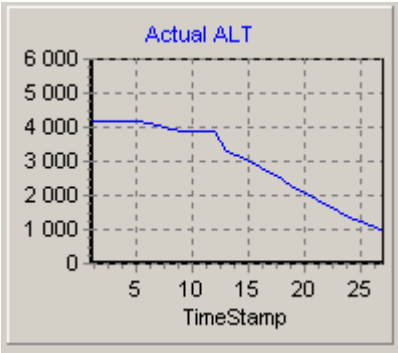
$$Kv = \frac{y_2 - y_1}{x_2 - x_1}, \tag{3}$$

$$T = \frac{t_0}{n}, \tag{4}$$

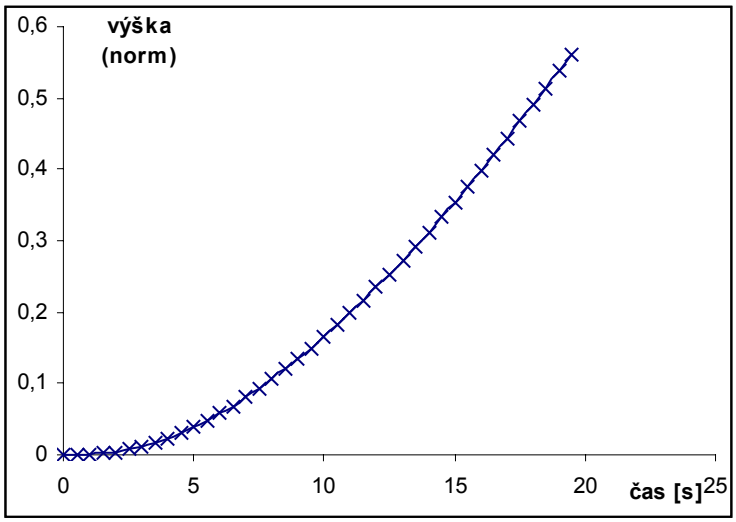
$$n = \frac{h(t_0)}{Kv}, \tag{5}$$

n	1	2	3	4	5	6
h(to)/kv	0.368	0.271	0.224	0.195	0.175	0.161

Tab. 7 závislost umístění inflexního bodu na řádu systému.



Obr. 42 přechodová charakteristika výšky modelu.



Obr. 43 normovaná přechodová charakteristika výšky modelu.

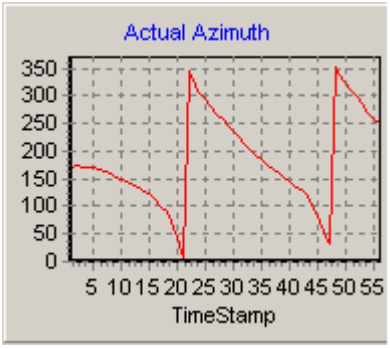
Na obr. 42 je přechodová charakteristika měřena od stropu laboratoře a vyznačuje pokles výšky (přiblížení ke stropu). Časová značka v grafu je stanovena na 800 *ms* a naměřené hodnoty výšky jsou uvedeny v *cm*. Na obr. 43 je charakteristika přepočtena a normována. Pomocí této přechodové charakteristiky byl vypočten aproximační přenos:

$$F(s) = \frac{0.08}{p(1+8p)}.$$

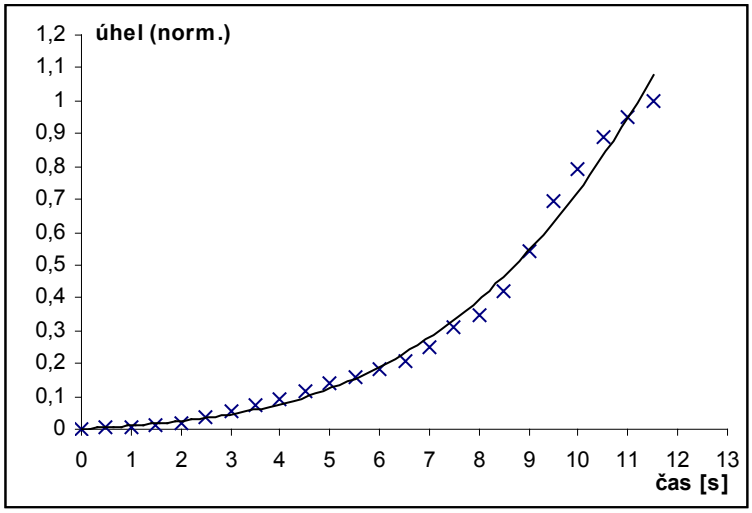
Natočení modelu

Dále byla změřena charakteristika natočení modelu v horizontální rovině v závislosti na skokové změně napětí na směrové vrtuli (obr. 44). Jak je z tohoto obrázku patrné, dochází ke skokové změně azimutu. Tato skoková změna je způsobena přechodem natočení modelu přes

severní stranu (rozsah 0-359 stupňů). Časová značka (TimeStamp) na obr. 44 je opět stanovena na 500 *ms*. Na obr. 45 je uvedena normovaná přechodová charakteristika.



Obr. 44 přechodová charakteristika natočení modelu.



Obr. 45 normovaná přechodová charakteristika natočení modelu.

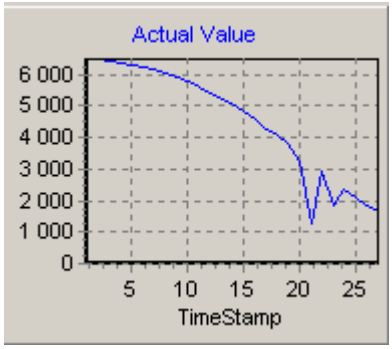
Z normované a přepočtené přechodové charakteristiky byl vypočten aproximační přenos podle stejného postupu jako v předchozím bodě, a je roven:

$$F(s) = \frac{0.1785}{p(1+6p)}.$$

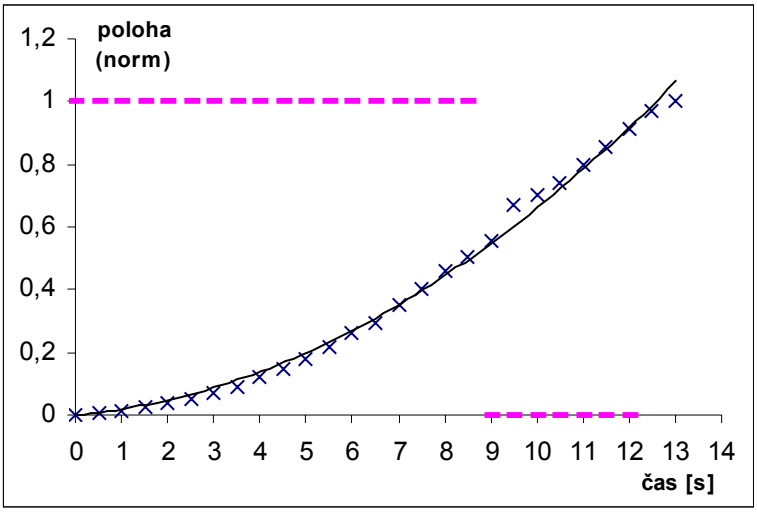
Poloha v horizontálním směru

Ke stanovení dynamických parametrů modelu ve směru horizontálního pohybu bylo rovněž využito přechodové charakteristiky. Při tomto měření bylo nutné zajistit vyvážní modelu, aby

byl vektor rychlosti ve směru tíhového zrychlení nulový. Přechodová charakteristika je uvedena na obr. 46. Časová značka na v grafu na obr.46 je rovna 500 *ms*.



Obr. 46 přechodová charakteristika polohy v horizontálním směru.

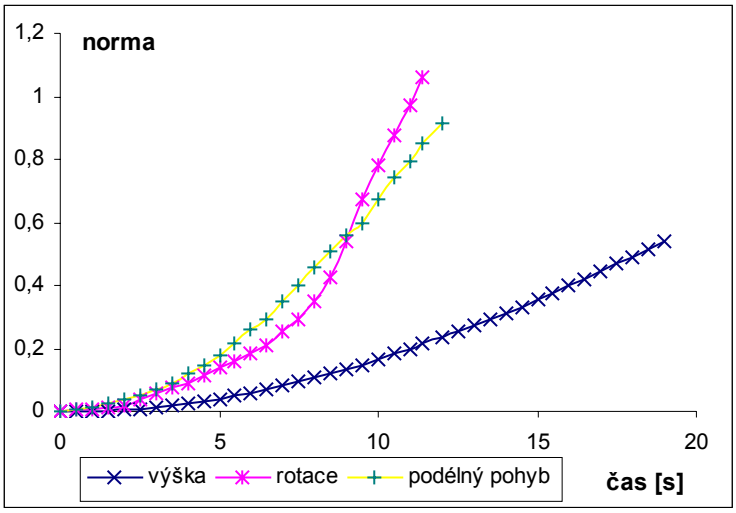


Obr. 47 přechodová charakteristika modelu v horizontálním směru.

Měření polohy probíhalo v závislosti na skokovém napětí hlavních pohonných motorů. Časový průběh vstupního napětí je označen v grafu normované charakteristiky čárkovaně (obr.47). Stanovení aproximačního přenosu bylo provedeno podle stejného postupu jako v předešlé části. Výsledný aproximační přenos je roven:

$$F(s) = \frac{0.1237}{p(1+4.5p)}.$$

Výsledné porovnání dynamických parametrů modelu je uveden na obr.48. Jde o výše uvedené normované přechodové charakteristiky.



Obr. 48 porovnání normovaných přechodových charakteristik.

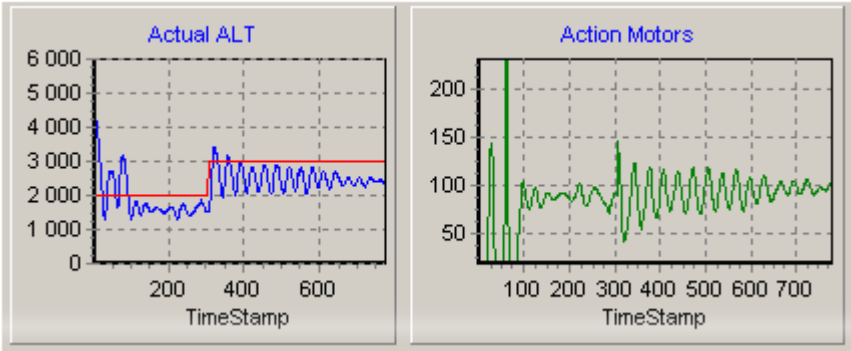
5.1.2 Návrh regulátoru

Regulace výšky

V této části bude popsán návrh regulátoru, pro aproximační přenos, který byl získán pomocí přechodové charakteristiky. Pro regulaci výšky byl zvolen regulátor typu PID, aby zajistil nulovou regulační odchylku a rychlou reakci na změnu výšky. Regulátor byl navržen pomocí GMK a výsledný přenos regulátoru je:

$$R(s) = r_1 \frac{[p^2 + (\omega_d + \varpi_i)p + \varpi_d \varpi_i]}{p} = \frac{r_1 p^2 + r_0 p + r_{-1}}{p} = \frac{0.021p^2 + 0.74 + 0.2}{p}.$$

Výsledek regulace je uveden na obr.49. Časová značka (TimeStamp) byla nastavena na 200 *ms*. Z obr. 49 je patrný kmitavý průběh regulace, který je způsoben nejen nepřesnou identifikací soustavy, ale převážně nelineárním řízením pohonných motorů.Nelineárním řízením je zde myšlena vzájemná závislost průběhu napětí motorů (PWM) na vstupním řízení obr.21 (nelineárním členem je tedy dekodér řídicího PWM viz kap. 3.1.1)

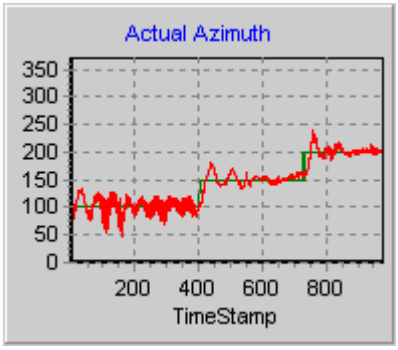


Obr. 49 výsledek regulace výšky.

Regulace směru

Pro regulaci této veličiny bylo také využito PID regulátoru.Výsledek této regulace je uveden na obr.50. Přechod přes nulovou úroveň byl řešen, monitorováním tohoto přechodu. V okamžiku, kdy došlo k přechodu přes severní stranu byla připočtena resp. odečtena hodnota 360 k aktuální hodnotě azimutu. Regulátor byl opět navržen pomocí metody GMK. Výsledný přenos regulátoru je:

$$R(s) = r_1 \frac{[p^2 + (\omega_d + \varpi_i)p + \varpi_d \varpi_i]}{p} = \frac{r_1 p^2 + r_0 p + r_{-1}}{p} = \frac{0.29p^2 + 2.4p + 1.6}{p}.$$

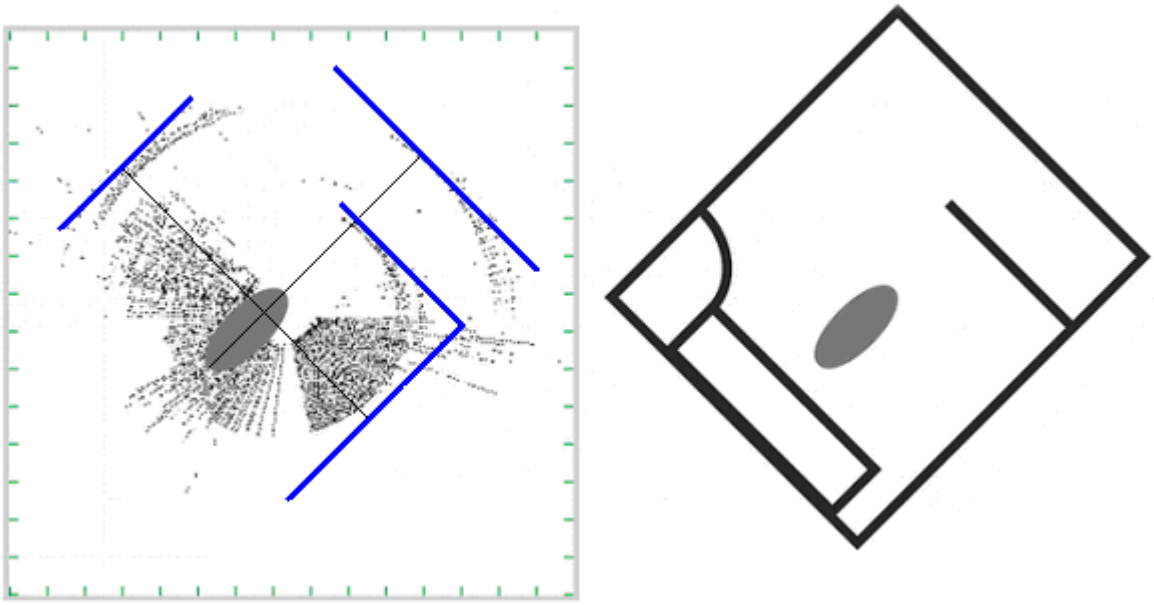


Obr. 50 regulace natočení modelu (odchylka od severu).

Jak je patrné z grafu na obr. 50 dochází při změně žádané hodnoty k překmitům, které jsou způsobeny stejnými vlastnostmi řízení jako v při řízení vertikální polohy modelu (nepřesná identifikace, nelineární převod akčního zásahu).

5.1.3 Využití naměřených dat k orientaci v prostředí

Tato část práce navazuje na diplomovou práci [3], která se zabývala řešením orientace v neznámém prostředí. Model vzducholodi je pro tuto orientaci vybaven otočným ultrazvukovým dálkoměrem obr. 20, umístěným vespod modelu. Pro lokalizaci volného prostoru je nutné zjistit pomocí otočných dálkoměrů obraz okolního prostoru a určit tak směr, který je volný. Aby obraz z otočných dálkoměrů nebyl zkreslený, musí být model v naprostém klidu. Protože zatím není k dispozici korektní identifikace systému a tím i korektně navržený regulátor, není možné tuto stabilní polohu zajistit. Možným řešením, které bylo aplikováno, je vytvoření obrazu okolí před odpoutání modelu od parkovacího doku. Příklad této mapy je uveden na obr. 51. Jak je patrné jde o řadu bodů, které reprezentují možné odrazy od překážek. Vzhledem k tomu, že tyto odrazy nemusí reprezentovat skutečnou překážku, mohou být například tvořeny násobnými odrazy mezi modelem a překážkou, není pro zpracování dat možné využít jednoduchý algoritmus. Na větší vzdálenosti (kolem 10 *m*) je poloměr kuželu, v kterém se šíří zvuková vlna značný (2,7 *m*) a neposkytne tak korektní představu o členitosti prostoru. Nedopadá-li zvuková vlna v kolmém směru na překážku, dochází k násobným odrazům (první jsou zachyceny nejkratší odrazy), které vytváří obraz „zakulacení“ rovných stěn. Bylo by tedy nutné naměřená data předzpracovat pomocí filtrování (váhováním několika vzorků ze stejného místa) a poté využít některé z dostupných metod pro rozpoznávání např. shluková analýza či zpracování obrazů např. detekce hran. Obraz „zakulacení“ rovných stěn lze následně nahradit tečnami ke změřenému obrazu v bodě s kolmým odrazem (nejsilnější odraz). Na obr. 52 je uveden reálně naměřený obraz okolí. Šedě je vyznačena vzducholoď. Barevně jsou naznačeny tečny, které nahrazují naměřený odraz „zakulacení“ rovných stěn. Obrázek vpravo uvádí skutečný obrys místnosti.



Obr. 51 výsledný obraz okolního prostředí.

6. Kapitola

Závěr

Tato práce se zabývá návrhem a realizací dálkového řízení mobilního zařízení s použitím modulů Bluetooth. Mobilní zařízení v tomto případě představoval reálný model vzducholodi, který je umístěn v laboratoři katedry řízení. Úkolem tedy bylo navrhnout a realizovat jak oblast týkající se hardwarové podpory tak část nutného programového vybavení, které zajistí komunikaci a zpracování naměřených dat ze senzorů pro orientaci v prostředí. S využitím těchto dat pak navrhnout řízení modelu v prostředí s překážkami. Část, která se věnovala návrhu a konstrukci řídicí jednotky, napájecí části a senzorů umístěných v modelu, byla předmětem paralelně realizované práce [3]. Šlo tedy o realizaci komplexní úlohy.

Řešení této práce je rozděleno do čtyřech hlavních kapitol. Základním cílem práce bylo vytvoření komunikačního bezdrátové spojení pomocí technologie Bluetooth. Popisem tohoto standardu se zabývá kapitola 2.

Následující kapitola je věnována popisu reálného modelu vzducholodi, jeho akčních členů, řídicí části a části pro komunikaci a modelem. Bezdrátové spojení zajišťoval vestavný modul SieMo od firmy Siemens. Poslední část této kapitoly je věnována popisu tohoto modulu.

Předposlední část práce se zabývá řešením softwarového vybavení, jak pro část komunikace, tak pro část řízení modelu. Tato kapitola pokrývá i programový návrh a realizaci programového prostředku pro navázání a přenos dat mezi dvěma osobními počítači (podle bodu 2 zadání). Tato část byla splněna a byla tak ověřena komunikace s mezi moduly SieMo.

Poslední částí zadaní bylo realizace řízení modelu v prostředí s překážkami. Tuto část nebylo možné plně realizovat, protože data získaná z ultrazvukových senzorů nedovolovala korektní interpretaci. Hlavní důvod, proč tato data nevyhovují pro řízení, je jejich velký rozptyl, způsobený mnohonásobnými odrazy a špatnou směrovou charakteristikou ultrazvukového senzoru. Pořízený obraz místnosti je zkreslený a nelze jej jednoduše využít pro řízení. Při pořizování obrazu místnosti není dále možné zajistit stabilní polohu modelu, protože není k dispozici regulační smyčka stabilizace polohy modelu. Jistým řešením by bylo

nevytvářet obraz celého prostoru, ale využít k detekci překážek pouze částečného natočení ultrazvukových dálkoměrů v kruhové výseči přibližně 30 stupňů před modelem. Omezení násobných odrazů by se dalo dosáhnout dálkového přenastavení citlivosti ultrazvukových dálkoměrů.

Dalším problémem realizace řízení modelu je nestálost parametrů tohoto modelu, které jsou ovlivňovány okolním prostředím (tlak, teplota). Možným řešením, které nevyžaduje klasický způsob identifikace parametrů modelu, je fuzzy řízení, které využívá k řízení definovaná pravidla a proměnnost parametrů modelu není pro řízení zásadní.

Jako nevhodné se ukázalo využití mikrokontroléru PIC 16F873 jako hlavní řídicí části modelu. Svým výkonem je tento procesor nevyhovující a měl by být nahrazen výkonnějším typem např. jednočipovým mikropočítačem Motorola HC12, který je vybaven větším počtem periférii Input Capture / Output Compare, které lze využít pro měření ultrazvukovými dálkoměry a pro generování PWM modulace. Dále mikropočítač dovoluje definovat různý dělící poměr pro generování přenosové rychlosti pro sériovou komunikaci a tak využít maximálního taktovacího kmitočtu.

Jelikož je model řízen pomocí bezdrátové technologie Bluetooth, která má definovaný standard IEEE, bylo by možné k manuálnímu řízení modelu využít například mobilního telefonu či PDA.

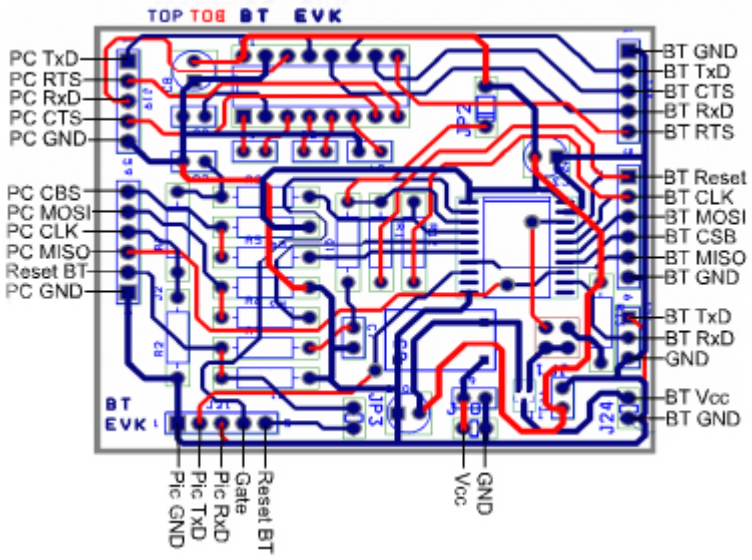
Dalším krokem při realizaci řízení by mohlo být využití jiného operačního systému s realtimovým jádrem, např. RT Linux.

Velikou výhodou by byla implementace přenášení HCI příkazů pomocí bezdrátového rozhraní a vytvořit tak dálkovou správu vzdáleného Bluetooth modulu.

Dodatek A

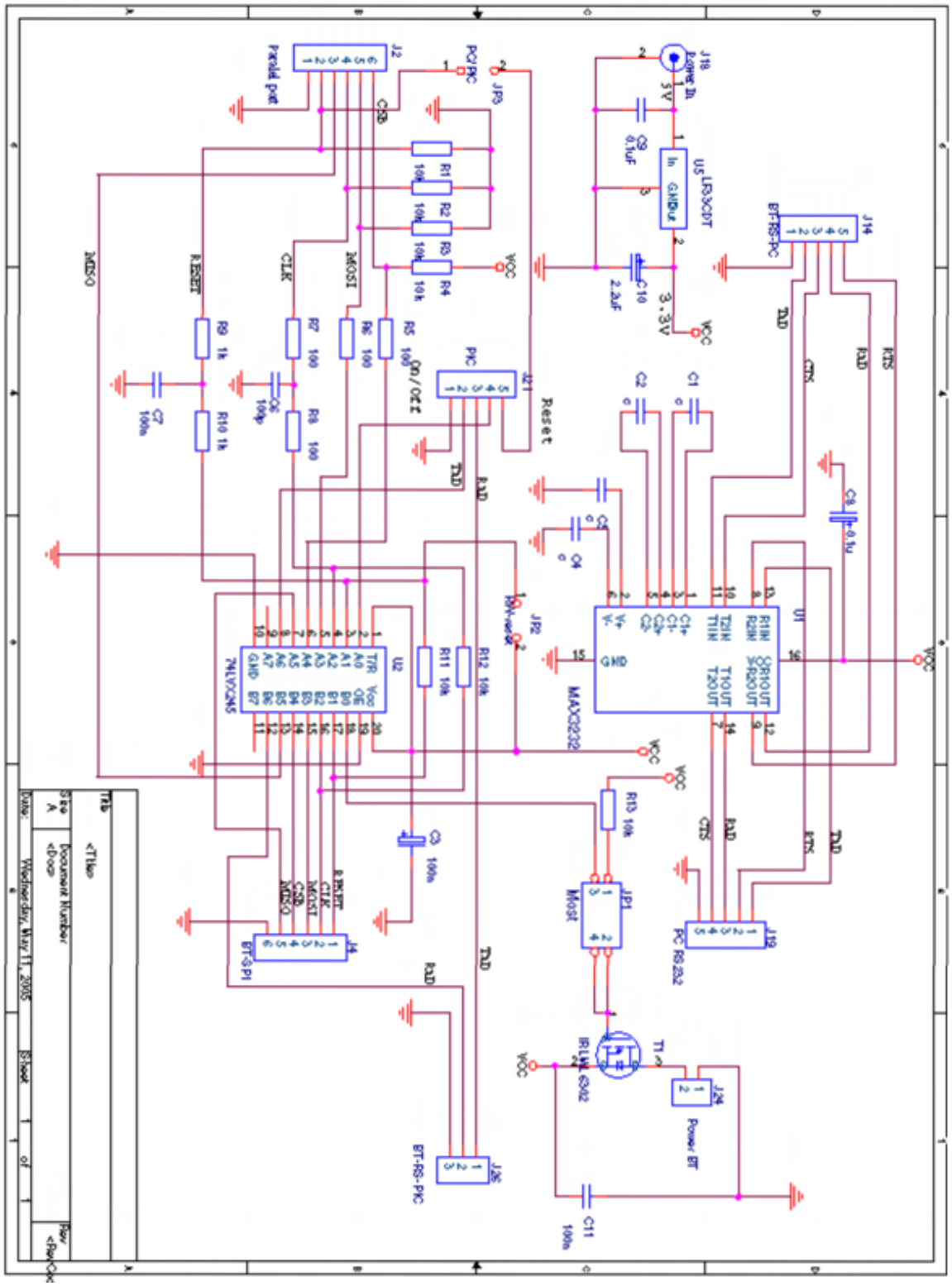
Schémata

A.1



Obr. A.1 obrazec plošných spojů s popisem vstupů a výstupů.

A.2



Obr. A.2 schéma zapojení vývojového modulu.

Dodatek B

technologie Bluetooth

B.1

Segment	TYPE code $b_3b_2b_1b_0$	Slot occupancy	SCO link	ACL link
1	0000	1	NULL	NULL
	0001	1	POLL	POLL
	0010	1	FHS	FHS
	0011	1	DM1	DM1
2	0100	1	undefined	DH1
	0101	1	HV1	undefined
	0110	1	HV2	undefined
	0111	1	HV3	undefined
	1000	1	DV	undefined
	1001	1	undefined	AUX1
3	1010	3	undefined	DM3
	1011	3	undefined	DH3
	1100	3	undefined	undefined
	1101	3	undefined	undefined
4	1110	5	undefined	DM5
	1111	5	undefined	DH5

Tab. B.1. typy paketů vrstvy Baseband.

B.2

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)
HV1	na	10	1/3	no	64.0
HV2	na	20	2/3	no	64.0
HV3	na	30	no	no	64.0
DV*	1 D	10+(0-9) D	2/3 D	yes D	64.0+57.6 D

Tab. B.2 parametry datových paketů SCO linky.

B.3

Type	Payload Header (bytes)	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate (kb/s)	Asymmetric Max. Rate (kb/s)	
						Forward	Reverse
DM1	1	0-17	2/3	yes	108.8	108.8	108.8
DH1	1	0-27	no	yes	172.8	172.8	172.8
DM3	2	0-121	2/3	yes	258.1	387.2	54.4
DH3	2	0-183	no	yes	390.4	585.6	86.4
DM5	2	0-224	2/3	yes	286.7	477.8	36.3
DH5	2	0-339	no	yes	433.9	723.2	57.6
AUX1	1	0-29	no	no	185.6	185.6	185.6

Tab. B.3 parametr datových paketů ACL linky.

B.4

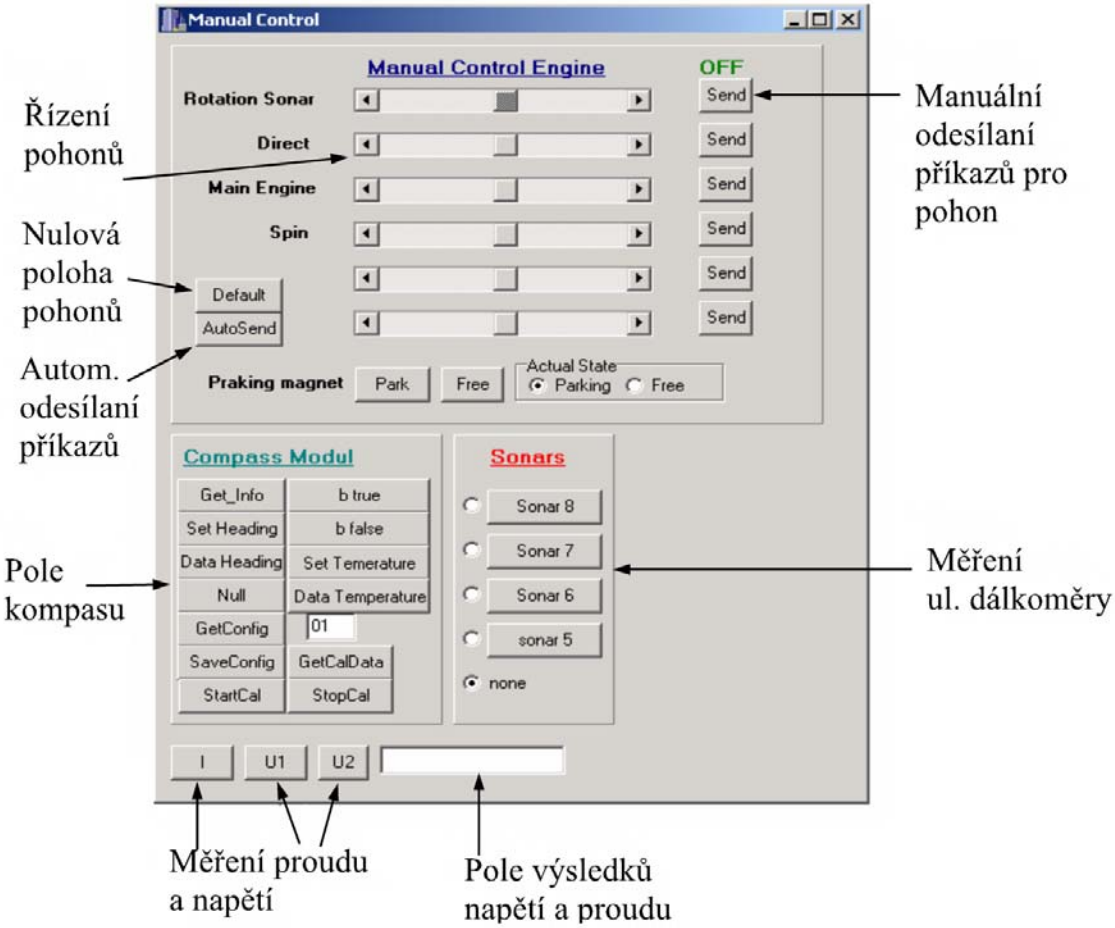
Type	User Payload (bytes)	FEC	CRC	Symmetric Max. Rate	Asymmetric Max. Rate
ID	na	na	na	na	na
NULL	na	na	na	na	na
POLL	na	na	na	na	na
FHS	18	2/3	yes	na	na

Tab.B.4 parametry řídicích paketů vrstvy Baseband.

Dodatek C

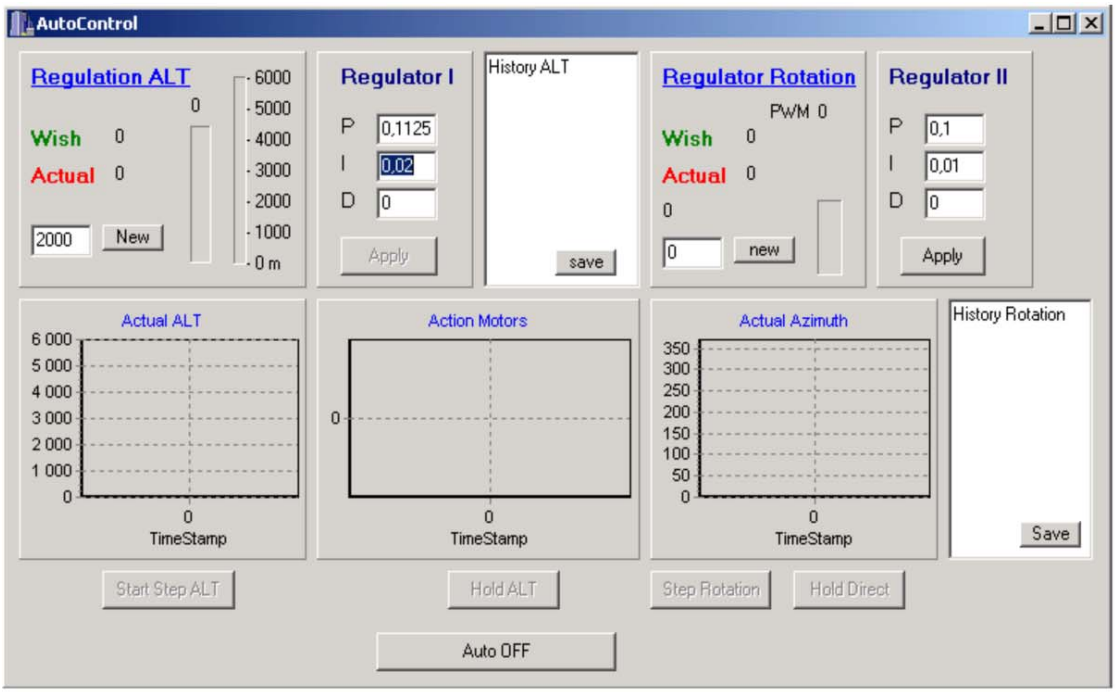
Popis programu

C.1



Obr. C.1 manuální řízení modelu.

C.2



Obr.C.2 automatické řízen modelu.

Reference

[1] *Specification of the Bluetooth System.* Volume 1, Verze 1.1. Bluetooth SIG, (PDF). Dostupné na http://www.Bluetooth.org/Spec/.

[2] MORROW, Robert. *BlueTooth operation and use*, McGraw-Hill 2002

[3] WOHLGEMUTH, Jan. (2005) *Diplomová práce Orientace v prostředí*.

[4] *SieMoS37005*, *DataBook*. Siemens (2002), (PDF)

[5] *BlueCore Serial Protocol,* CSR, [online], (2005) http://www.csr.com/pr/pr058.htm

[6] *Persistent Store Key Settings*. CSR (2001), (PDF)

[7] *Bezdrátové komunikace v automatizační parxi II: standard bluetooth* [online], 2005 http://www.automa.cz/automa/2003/au070338.htm/.

[8] *Standarad: Bluetooth technologie* [online], 2005 http://www.opencae.cz/hw/hw_standard_bluetooth.html

[9] *PNI Corp. V2Xe*, elektronický kompas tech. specifikace (PDF)

[10] *PIC 16F87x Data Sheet*, Microchip Technology inc, 2001 (PDF).

[11] VACEK, Václav. *Sériová komunikace ve WIN*, 1. vydaní, BEN, Praha 2003

[12] *Bluetooth Tutorial – Specifications* [online], 2005 http://www.palowireless.com/infotooth/tutorial.asp/.

[13] *Bezdrátové komunikace v automatizační parxi I:historie a současnost* [online], 2005, http://www.automa.cz/automa/2003/au050335.htm/.

[14] BURKHARD, Kainka. Využití rozhraní PC, 1. vydání, HEL, Ostrava

[15] VYSOKÝ, Ondřej. *Elektronické systémy 2*, 2. vydání, ČVUT, Praha 2003

[16] JOHN, Jan. *Systémy a řízení*, 1. vydání, ČVUT, PRAHA 1999

[17] HORÁČEK, Petr a FUKA, Jindřich. *Systémy a modely*, 2. vydání, ČVUT, PRAHA 2001

[18] ŠTECHA, Jan a HAVLENA, Vladimír. *Teorie dynamických systémů*, 2. vydání ČVUT, PRAHA, 2002

Obsah CD

/DOC – offline dokumentace využitá k řešení práce.

/DP – diplomová práce ve formátu PDF a DOC.

/APL – aplikace využité k řešení diplomové práce.

/SPIC – zdrojové kódy pro mikrokontrolér.

/SCPPZ – zdrojové kódy programu pro řízení modelu pro osobní počítač.

/SCPPC – zdrojové kódy programu pro komunikaci mezi osobními počítači.