

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ROBOTICKÉ VOZÍTKO ŘÍZENÉ S VYUŽITÍM PLATFORMY ARDUINO

BAKALÁŘSKÁ PRÁCE

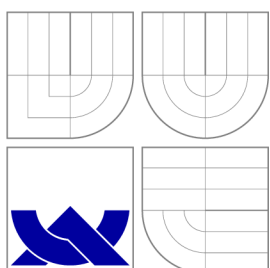
BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN ČECHMÁNEK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ROBOTICKÉ VOZÍTKO ŘÍZENÉ S VYUŽITÍM PLATFORMY ARDUINO

ROBOTIC ROVER CONTROLLED WITH USING THE ARDUINO PLATFORM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN ČECHMÁNEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN SAMEK, Ph.D.

BRNO 2013

Abstrakt

Cílem této práce je představit možnosti bezdrátového řízení robotů s využitím platformy Arduino a popsat vlastnosti, možnosti a využití této platformy. V práci je implementován obousměrný bezdrátový přenos mezi moduly pomocí XBee modulů. Praktická práce se skládá ze dvou částí - modulu ovladače a modulu robotického vozítka. Modul ovladače slouží pro řízení druhého modulu a obsahuje zobrazovací, signalizační a ovládací prvky. Modul vozítka, jehož základem je robot SRV-1, obsahuje ultrazvukové senzory pro měření vzdálenosti mezi vozítkem a překážkou v jeho okolí, a LED světelnou rampu pro osvětlení prostoru mimo směr jízdy vozítka.

Abstract

The goal of this bachelor's thesis is to introduce options of wireless robots control with using Arduino platform and describe preferences, options and use of this platform. In this thesis is implemented duplex wireless transfer between modules with XBee modules. Practical part of thesis consists of 2 parts - control module and robotic rover module. Control module is used for controlling the other module and it contains display, signal and control parts. Rover module, which is based on SRV-1 robot, contains ultrasonic sensors for distance measuring between rover and barrier around its place, and LED light ramp for lighting places out of the current direction of movement.

Klíčová slova

Arduino, robotické vozítko, LCD displej, XBee, joystick, Surveyor

Keywords

Arduino, robotic rover, LCD display, XBee, joystick, Surveyor

Citace

Martin Čechmánek: Robotické vozítko řízené s využitím platformy Arduino, bakalářská práce, Brno, FIT VUT v Brně, 2013

Robotické vozítko řízené s využitím platformy Arduino

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Samka, Ph.D.

.....
Martin Čechmánek
5. května 2013

Poděkování

Chtěl bych poděkovat vedoucímu této práce Ing. Janu Samkovi, Ph.D. za poskytnutou pomoc a rady, Ing. Václavu Šimkovi za pomoc při vytváření plošných spojů, své rodině a všem, kteří mě v průběhu tvorby práce podporovali.

© Martin Čechmánek, 2013.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	4
1.1 Cíl práce	4
2 Platforma Arduino a bezdrátová komunikace	5
2.1 Popis platformy	5
2.2 Vývojové prostředí	5
2.3 HW Arduino a jeho druhy	6
2.4 Možnosti rozšíření	8
2.5 Bezdrátové moduly	8
2.5.1 WiFi	8
2.5.2 Bluetooth	9
2.5.3 XBee	9
3 Možnosti řízení robotického vozítka pomocí platformy Arduino	12
3.1 Robotické vozítko	12
3.2 Ovládací prvky	13
3.3 Zobrazovací prvky	13
3.4 Komunikační kanál	13
3.5 Detektory a jiné snímače	14
3.6 Autonomní ovládání	14
4 Návrh a implementace	15
4.1 Návrh komunikačního protokolu	15
4.2 Zvolené řešení	16
4.3 Návrh modulů	17
4.4 Použité periferie	17
4.4.1 Zobrazovače	17
4.4.2 Řídicí prvky	18
4.4.3 Detektory a snímače	18
4.4.4 Konstrukční prvky	18
4.4.5 Světelná rampa	19
4.4.6 Motory	20
4.5 Modul ovladače	20
4.6 Modul vozítka	21
4.7 Řídicí program	23
4.7.1 Použité knihovny	23
4.7.2 Inicializace	24
4.7.3 Režim párování	25

4.7.4	Režim přenosu	26
4.7.5	Režim příjmu	28
4.7.6	Autonomní režim	29
4.8	Zhodnocení a testování	30
5	Závěr	32
A	Obsah CD	34
B	Zdrojové kódy	35

Seznam obrázků

2.1	Vývojové prostředí pro platformu Arduino.	6
2.2	Přehled druhů desek platformy Arduino: 1. LilyPad, 2. UNO, 3. Fio, 4. Pro mini, 5. Mini, 6. Nano, 7. Mega 2560, 8. Uno, 9. Wiring board, 10. Pro, 11. Ethernet, 12. Bluetooth, 13. Duemilanove. Obrázek byl převzat z knihy [7].	7
2.3	Moduly pro Arduino rozšiřující o dotykový displej, Ethernet rozhraní, či GPS modul. Obrázky převzaty ze stránek [12].	8
2.4	Obrázek XBee modulu Série 1 (vlevo) a Série 2.	10
2.5	Úvodní obrazovka a možnosti nastavení XBee modulu v programu X-CTU.	11
2.6	Modul XBee Explorer pro připojení konfigurátoru XBee modulů k PC přes USB rozhraní.	11
3.1	Robotické vozítko SRV-1 od společnosti Surveyor [1].	13
3.2	Robot sledující čáru PICAXE – 20X2 Microbot. Obrázek převzat z http://www.picaxe.com/Hardware/Robot-Kits/PICAXE-20X2-Microbot/	14
4.1	Diagram komunikace modulů při zasílání zpráv.	16
4.2	Ultrazvukový senzor vzdálenosti HC – SR04.	18
4.3	Schématické zapojení s použitím optočlenu KB847. Vlevo zapojení pro ovládnání periferního zařízení (výstupní vývod Arduino), vpravo pak zapojení pro čtení hodnoty ze zařízení (vstupní vývod).	18
4.4	Umístění modulu ovladače do konstrukční krabičky U-KP18.	19
4.5	Detail světelné rampy robotického vozítka.	19
4.6	Modul ovladače - čelní pohled. Čísla komponent spolu s jejich funkcí jsou uvedena v Tabulce 4.2.	20
4.7	Schématické zapojení rozšiřující desky pro Arduino UNO.	21
4.8	Schématické zapojení rozšiřující desky pro Arduino FIO.	22
4.9	Schématické zapojení rozšiřujících desek pro modul ovladače (Arduino UNO – vlevo) a modul vozítka (Arduino FIO – vpravo).	22
4.10	Konstrukce světelné rampy robotického vozítka.	23
4.11	Diagram inicializační části programu obou modulů.	24
4.12	Diagram průběhu komunikace při párování modulů.	25
4.13	Diagram programu modulu ovladače.	27
4.14	Ukázka zobrazovacích režimů LCD displeje u modulu ovladače.	27
4.15	Diagram programu modulu vozítka.	28
4.16	Pseudokód algoritmu pro řízení robota v autonomním režimu.	30
4.17	Výsledná podoba modulu robotického vozítka.	31
4.18	Detailní pohled na vnitřní usazení jednotlivých komponent modulu vozítka.	31

Kapitola 1

Úvod

Při volbě zadání bakalářské práce jsem zvažoval mé možnosti i směr zájmu. Jelikož se zajímám o roboty a robotické řízení, již za svou praktickou maturitní práci jsem si zvolil z této oblasti téma s názvem *Návrh konstrukce manipulátoru*, v které jsem navrhl a následně vyrobil manipulátor ze stavebnice Merkur ovládaný procesorem AT89S52 a poháněn unipolárními krokovými motory. Kvůli svému zájmu v této problematice jsem se rozhodl pro bakalářskou práci vyzkoušet (pro mne) novou platformu, která by měla příjemné uživatelské prostředí a zároveň jednoduchou syntax programovacího jazyka. To vše s využitím připojení přes sériový port USB, který je v dnešní době velmi rozšířený a starší platformy poskytují pouze (dnes již zastaralé) porty RS232 či LPT.

1.1 Cíl práce

Cílem této práce je představit možnosti bezdrátového řízení robotů s využitím platformy Arduino a samotné možnosti této platformy. Toto je popsáno v úvodní kapitole, spolu s druhy desek Arduino, možnostmi jejího rozšíření o přídavné moduly a také je zmíněno vývojové prostředí, ve kterém lze psát zdrojové kódy programů. Také jsou zde analyzovány možnosti a typy bezdrátové komunikace zařízení, včetně ukázky a porovnání parametrů modulů pro bezdrátovou komunikaci.

V kapitole 3 je proveden rozbor možných řešení při řízení robotů a podobných zařízení z hlediska HW prostředků. Jsou zde nastíněny varianty periferních zařízení, které se běžně používají na robotech a slouží pro jejich řízení, orientaci, komunikaci a informování o svém stavu či stavu jejich okolí.

Kapitola s názvem *Návrh a implementace* popisuje konkrétní podobu praktické části této práce. V úvodu této kapitoly je nastíněn postup návrhu jak programové části spolu komunikačním protokolem, tak i výběr komponent, včetně návrhu plošných spojů pro oba moduly. Detailně je popsán rozbor komunikace mezi moduly, stejně jako celého řídicího programu obou modulů.

Závěr obsahuje shrnutí práce, splnění cílů a úvahu nad možnými rozšířeními navrženého zařízení. Také je zde vyzvednut vlastní přínos i existující nedostatky praktické práce.

Kapitola 2

Platforma Arduino a bezdrátová komunikace

Arduino je open-source platforma pro vytváření elektronických prototypů, založená na flexibilitě, jednoduše použitelném hardwaru a softwaru [2].

2.1 Popis platformy

Platforma Arduino byla navržena především pro všechny ty, kteří nemají zkušenosti s elektronikou či programováním softwaru. Díky dostupnosti návodů a příkladů pro tuto platformu je snadné i pro netechnické typy snadno vytvářet malé i velké projekty, od rozsvěcování LED diody, přes ovládání motorů, až třeba k autonomnímu ovládání robotů či vytváření bezdrátových sensorových sítí. Tato platforma je uplatnitelná v širokém spektru použití, může být využita pro výukové účely a díky připojení k PC přes USB lze desky Arduino programovat v jakémkoli zařízení s tímto portem.

Další výhodou pro nezkušené uživatele je to, že s touto platformou dokáží velmi jednoduše pomocí nepájivého kontaktního pole a základních součástek vytvořit jednoduchý projekt a otestovat si pochopení principů programování tohoto hardwaru. To vše díky konektorům vyvedených po okrajích desky, do kterých lze připojit různá rozšíření, součástky apod. [10].

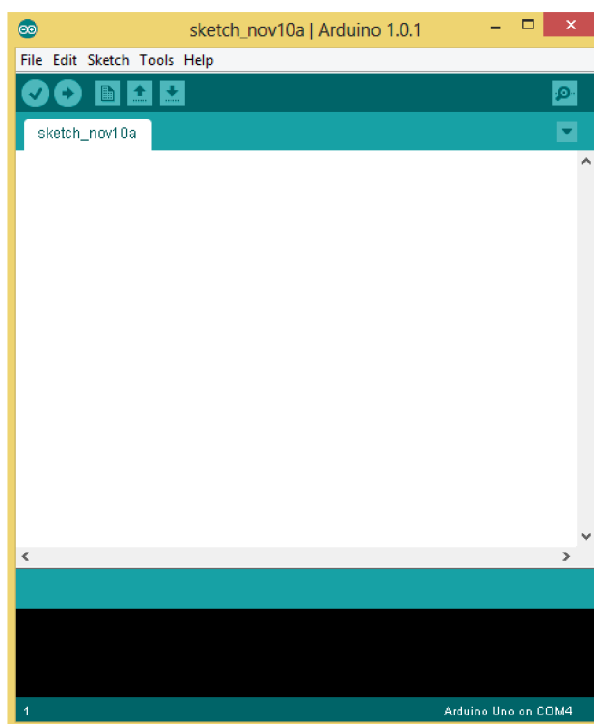
2.2 Vývojové prostředí

Vývojové prostředí pro aplikace platformy Arduino se běžně vytvářejí v grafickém uživatelském prostředí Arduino IDE. Tento open-source software je volně ke stažení na oficiálních stránkách Arduino <http://www.arduino.cc> [2] pro Windows, Mac i Linux. Pro projekty v tomto prostředí se užívá názvu „sketch“, což přeloženo do češtiny znamená skeč, skica, náčrt. Po spuštění této aplikace se tedy otevře prázdný skeč, do kterého již můžeme psát svůj vlastní program, jak lze vidět na Obrázku 2.1.

Jak již bylo zmíněno, Arduino nabízí předem připravené skeče pro ukázkovou činnost různých zařízení. Lze je vybrat z menu *File*→*Examples*, takže pokud má uživatel k dispozici desku s Arduinem a patřičné periferie k projektu, může si je přeložit, naprogramovat do zařízení a rovnou otestovat. To vše nabízí tento software vyvinutý pro Arduino, spolu s dalšími funkcemi jako jsou ukládání, nahrávání či vytváření nových projektů, zmiňovaná kompilace a naprogramování zařízení a v neposlední řadě také sledování komunikace sériové

linky (v případě že v naší aplikaci využíváme sériový přenos). Pokud již máme uloženy některé své skeče, snadno je zobrazíme v menu *File*→*Sketchbook*. Záložka *Tools* nabízí kromě automatického formátování textu také nastavení programátoru, kterým bude program nahrán do zařízení, samotný typ tohoto zařízení a sériový port, ke kterému je toto zařízení připojeno.

Programovací jazyk, v kterém jsou skeče pro Arduino psány, vychází z projektu **Wiring** [2] a je založen na jazyce C++. Soubor s kódem pro Arduino je uložen s příponou *.pde*, a lze jej editovat v libovolném textovém editoru, nicméně prostředí Arduino IDE nabízí zvýrazňování syntaxe i pohodlnou správu vytvořených programů. Je také možné u každého projektu nastavit klíčová slova, která se budou zvýrazňovat spolu s ostatními klíčovými slovy.



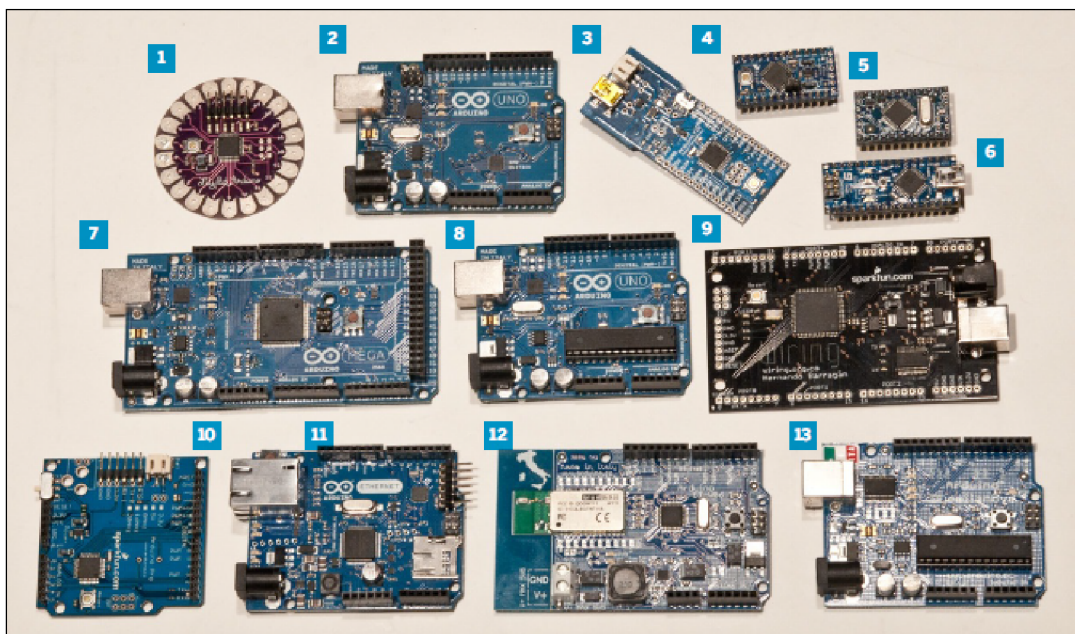
Obrázek 2.1: Vývojové prostředí pro platformu Arduino.

2.3 HW Arduino a jeho druhy

K tomu, abychom svůj program mohli spustit a testovat, je zapotřebí desky Arduino. Na Obrázku 2.2 je přehled různých typů HW pro vyvíjení aplikací pro tuto platformu. Každé se liší počtem vstup/výstupních vývodů, velikostí paměti, svými rozměry i využitím. Existují i specificky zaměřené typy desek jako je například Arduino Ethernet, který usnadňuje práci při programování služeb přes Ethernet rozhraní.

Existují oficiální desky Arduino, ale i řada kompatibilních desek od jiných výrobců [10] a s tím související i rozšíření pro tyto desky, jak je zmíněno později. Pro každý druh Arduino desky je potřeba v prostředí Arduino IDE nastavit patřičný typ, kvůli nastavení programovacích parametrů. Převážná většina desek je osazena mikrokontroléry Atmel, které tvoří srdce desky Arduino, a jejich vývody jsou vyvedeny na řadové konektory k okrajům

desky. Na typu mikrokontroléru také záleží zmiňované vlastnosti a tím i celá koncepce desky Arduino. Ve FLASH paměti mikrokontroléru se nachází *bootloader*, který umožňuje programovat Arduino bez externího programátoru [10].



Obrázek 2.2: Přehled druhů desek platformy Arduino: 1. LilyPad, 2. UNO, 3. Fio, 4. Pro mini, 5. Mini, 6. Nano, 7. Mega 2560, 8. Uno, 9. Wiring board, 10. Pro, 11. Ethernet, 12. Bluetooth, 13. Duemilanove. Obrázek byl převzat z knihy [7].

Napájení desky je možné přes USB port, který zároveň slouží jako spojení s PC pro programování mikrokontroléru, případně využití externí baterie (např. u Arduino FIO). Většina typů pracuje s napětím 5 V, jsou však takové, které poskytují na výstupu svých I/O pinů 3.3 V (např. FIO). Počet pinů se také různí, nicméně existuje několik typů použití těchto pinů:

- **Digitální vstup/výstup.**
- **PWM.**
- **Analogový vstup/výstup.**
- **Napájecí vstupy/výstupy.**
- **Resetovací vstup.**

Digitální vstupy/výstupy slouží, jak již název napovídá, k ovládní zařízení pomocí logických hodnot 0 (LOW) a 1 (HIGH). Těchto vývodů bývá na desce většina, jelikož je pravdivostní ovládní (zapnuto/vypnuto) nejčastěji využíváno k řízení připojitelných periférií. **PWM** je označení pro ovládní pulzně šířkovou modulací, což znamená, že změnou poměru zapnuto/vypnuto v pevně dané periodě opakování je měněna hodnota napětí na výstupu. Tato technika emuluje D/A převodník a užívá se například pro regulaci svitu LED diod, ovládní otáček stejnosměrných motorů apod. Tyto výstupy jsou většinou uváděny jak digitální V/V s tím, že je lze používat jak pro PWM, tak pro klasickou funkci digitálního

V/V. Do digitálních vstupů/výstupů se řadí i vývody pro přenos sériové komunikace (Tx, Dx, aj.). **Analogové vstupy/výstupy** mají své vlastní místo na desce a slouží ke snímání hodnot napětí v rozmezí $0\text{ V} - U_{cc}$, tedy běžně $0 - 5\text{ V}$, v programu následně interpretována jako číselná hodnota v intervalu $0 - 1024$. **Napájecí piny** jsou vyvedeny taktéž, poskytují pracovní úroveň napětí (5 V , 3.3 V atd.), zemnicí úroveň (GND) a vstupní napětí zdroje (Aref). **Resetovací pin** slouží pro ruční reset mikrokontroléru na desce.

2.4 Možnosti rozšíření

Jelikož se jedná o platformu vhodnou pro výuku či vývoj, je pro ni nachystána velká škála rozšíření. Tato rozšíření se na Arduino připojují pomocí tzv. *shieldů*¹ a umožňují Arduino vybavit různými zařízeními - například modulem dotykového displeje, rozhraním pro Ethernet, nebo modulem GPS, jak lze vidět na Obrázku 2.3. Tyto moduly jsou navrženy převážně pro verzi Arduino UNO (a vývodově kompatibilní), nicméně existují i redukce např. z Arduino UNO na Arduino MEGA. Po zasunutí modulu do desky je v převážné většině případů možné i nadále využívat všech pinů Arduino díky vyvedeným řadovým konektorům na modulu.



Obrázek 2.3: Moduly pro Arduino rozšiřující o dotykový displej, Ethernet rozhraní, či GPS modul. Obrázky převzaty ze stránek [12].

2.5 Bezdrátové moduly

Nutnost bezdrátového spojení je spojena s požadavky na volný pohyb vysílače a přijímače bez fyzického spojení. Řešení tohoto problému umožňují právě bezdrátové moduly, které v závislosti na svých vlastnostech zprostředkovávají komunikaci bez nutnosti přímého spojení obou zařízení. Příklady jednotlivých druhů jsou popsány níže.

2.5.1 WiFi

Standard WiFi nabízí širokopásmové bezdrátové připojení, které je v současné době velmi rozšířené (v domácnostech, ve školách, v podnicích, veřejné přístupové body aj.). Toto připojení komunikuje na nelicencovaných pásmech rádiového vysílání, což znamená, že zařízení

¹Shield je označení desky plošných spojů s určitým zařízením či rozšířením pro Arduino. Zasouvá se do stávajících portů desky a tytéž distribuuje dále, přičemž některé z nich (i všechny) využívá pro rozšiřující funkci. Přehled nabízených rozšíření seřazených podle výrobce je na [12].

mimo jiné musí dodržovat výkonové limity a cena těchto zařízení je vysoká. Obecně se dá říci, že použití WiFi brány je dražší a méně spolehlivé řešení než použití např. rozhraní Ethernet, nicméně to může být pro danou lokalitu jediná možnost [6].

Také pro platformu Arduino existují moduly rozšíření s rozhraním WiFi, nicméně fakulta tyto moduly nevlastní a tudíž je ani ve vlastní práci nebudu využívat.

2.5.2 Bluetooth

Bluetooth je komunikační protokol navržený k náhradě drátových připojení různých zařízení. Podle typu zařízení je také volen tzv. *profil* - například pro prostou sériovou komunikaci slouží profil **SPP** (Serial Port Profile), bezdrátová sluchátka využívají **Headset profile**, bezdrátová myš či klávesnice zase **HID** (Human Interface Device) profil atd. Komunikace probíhá přes 2 datové vodiče, pojmenované **Tx** a **Rx**, které zajišťují asynchronní sériový přenos. Tento protokol je hojně využíván pro malé bezdrátové sítě s krátkým dosahem [7].

Moduly pro Bluetooth jsou také dostupné na internetovém obchodu firmy **SparkFun.com**, nicméně jsem se nerozhodl ani pro tento protokol bezdrátové komunikace v mé práci.

2.5.3 XBee

Xbee je bezdrátový komunikační protokol založený na protokolu ZigBee (IEEE 802.15.4), pracující na frekvenci 2.4 GHz. Tento protokol je pak využíván ve všech XBee zařízeních, které jsou hlavně díky nízké ceně velmi rozšířené nejen pro využití s platformou Arduino [10].

Popis rozhraní

XBee moduly, které využívám ve své práci, jsou zapůjčeny ze školních prostředků. Tyto moduly byly zakoupeny od společnosti **SparkFun Electronics**, která se zabývá prodejem elektronických součástek a modulů pro platformy Arduino a jiné. Moduly, jejichž výrobce je firma **MaxStream Inc.**, jsou pak pro využití s deskami Arduino připojeny přes *shieldy*, které **SparkFun** taktéž nabízí.

Verze a druhy

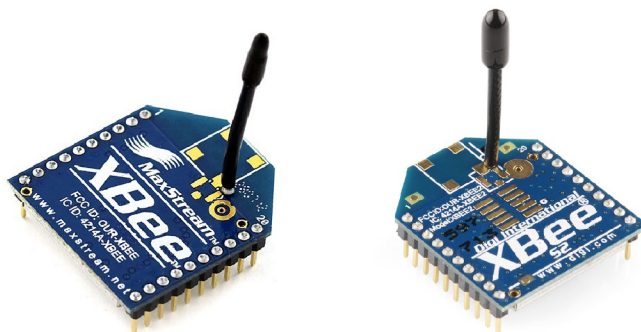
Existují 2 základní verze těchto modulů: **Série 1** a **Série 2**. Tyto verze se liší ve spotřebě, dosahu (odvíjející se od použité antény), ale hlavním rozdílem je komunikační protokol a s tím související používaná topologie zařízení v síti [8]. Jednotlivé vlastnosti těchto verzí jsou popsány níže. Z hlediska požadavků jsem zvolil ke své implementaci moduly Série 1, které sice nenabízí pokročilé možnosti komunikace a spojení, nicméně tyto vlastnosti v mé implementaci nejsou potřeba.

Série 1

Moduly první série jsou určeny především pro *point-to-point* komunikaci případně *point-to-multipoint*. Jsou také snazší pro řízení, nabízí 8 digitálních vstupů/výstupů, jejich dosah je ve vnitřních prostorech cca 30 metrů, ve venkovním prostředí až 100 metrů. Rozšířená verze *PRO* zvyšuje dosah na 100/300 m [6].

Série 2

Tato verze je vhodná pro vytváření sensorových sítí s mnoha zařízeními v síti a vyžaduje složitější konfiguraci. I když je rozložení vývodů Série 1 a Série 2 stejné, tyto mezi sebou nedokáží komunikovat [5]. Moduly obou verzí jsou vyobrazeny na Obrázku 2.4.



Obrázek 2.4: Obrázek XBee modulu Série 1 (vlevo) a Série 2.

Tabulkové srovnání těchto hodnot je uvedeno v Tabulce 2.1, převzato z [6].

	Série 1	Série 2
Typický dosah	30 m	40 m
Nejlepší možný dosah	100 m	120 m
Proud při odesílání/přijímání	45/50 mA	40/40 mA
Firmware	802.15.4 point-to-point	ZB ZigBee mesh
Digitální I/O piny	8 (plus 1 pouze vstupní)	11
Analogové vstupy	7	4
Analogové výstupy (PWM)	2	–
Point-to-point, hvězdicová topologie	Ano	Ano
Mesh síť	Ne	Ano
Jediný firmware pro všechny uzly	Ano	Ne
Vyžadován koordinační uzel	Ne	Ano
Konfigurace point-to-point	Jednoduchá	Složitější
Výrobce čipsetu	Freescale	Ember
Dostupný firmware	802.15.4 (IEEE standard)	ZB (ZigBee 2007)
Aktivně podporován	Ano	Ano

Tabulka 2.1: Srovnání parametrů XBee modulů Série 1 a Série 2.

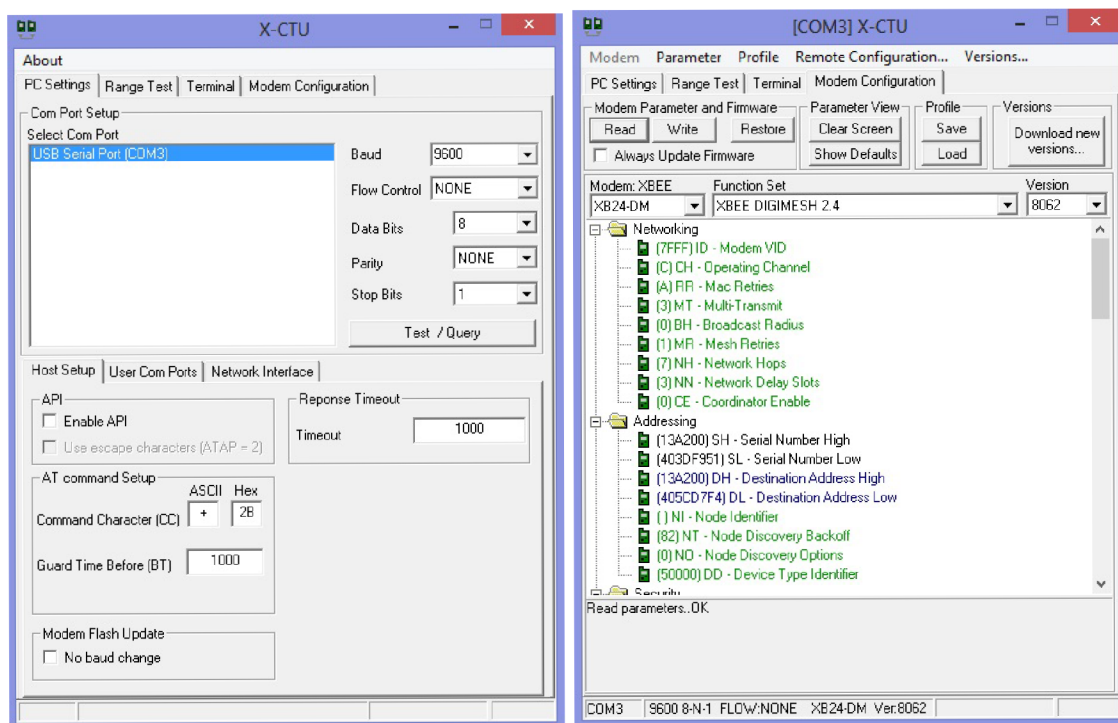
Konfigurace

Použité moduly lze nakonfigurovat pro potřeby uživatele pomocí softwarových nástrojů (např. X-CTU), či pomocí samotné desky Arduino s nastavením požadovaných propojek [3]. Aplikace X-CTU po jejím spuštění zjistí zařízení připojená na sériové porty počítače, přičemž nabízí test zařízení na tomto portu (pro kontrolu funkčnosti spojení). Pokud test uspěl, lze přistoupit k nastavení.

Modul XBee při konfiguraci přes aplikaci X-CTU vyžaduje modul XBee Explorer, jež je vyobrazen na Obrázku 2.6. Na připravené řadové konektory se dle popisku na desce plošných

spojů připojí bezdrátový modul XBee. Po připojení k PC přes USB kabel a spuštění X-CTU aplikace již nezbyvá než přikročit k testu komunikace (LED diody na XBee Exploreru by měly signalizovat přenos) a jeho následné konfiguraci.

Obrázek 2.5 znázorňuje parametry každého XBee modulu, zde konkrétně verze Série 1. Barevně rozlišené řádky udávají význam jednotlivým parametrům, např. černé řádky jsou nezměnitelné (MAC adresa zařízení), zelené řádky editovatelný atribut atp.



Obrázek 2.5: Úvodní obrazovka a možnosti nastavení XBee modulu v programu X-CTU.



Obrázek 2.6: Modul XBee Explorer pro připojení konfigurátoru XBee modulů k PC přes USB rozhraní.

Kapitola 3

Možnosti řízení robotického vozítka pomocí platformy Arduino

Pro svou práci jsem se rozhodoval mezi různými koncepcemi návrhu řízení robotického vozítka. Obecně jednou z variant je řízení přes PC, ke kterému je připojen jeden z modulů Arduino a uživatel přes PC řídí robota. Ovládání může být implementováno softwarově (uživatel ovládá aplikaci s prvky pro řízení zařízení), případně pomocí klávesnice. Toto řízení skýtá značnou náročnost na implementaci, nutnost znalosti nejen programovacího jazyka pro samotné řízení robota, ale také znalost programování aplikací pro komunikaci a ovládání zařízení přes PC.

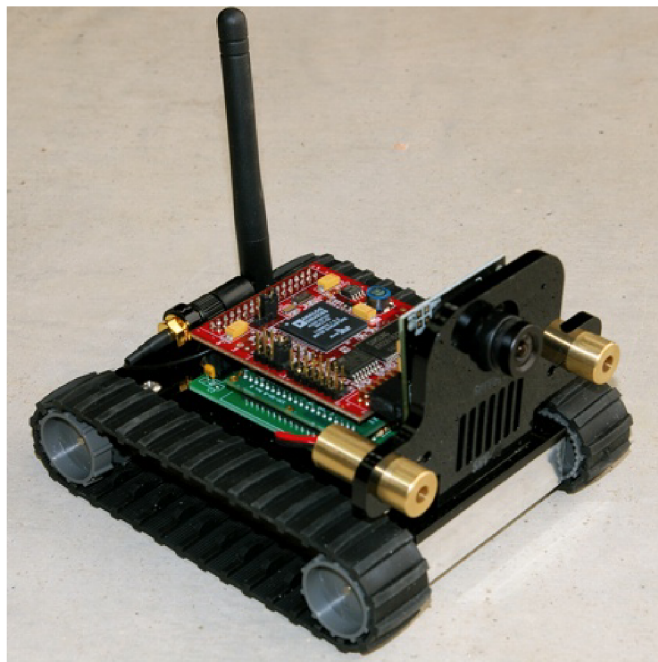
Zařízení lze také ovládat řídicími prvky umístěnými přímo na zařízení (případně mohou být tyto prvky s robotem propojené kabelem) - zde není potřeba druhého ovládacího členu, jelikož jsou řídicí příkazy zpracovávány pouze modulem umístěným na ovládaném zařízení. Toto řízení vyžaduje nejmenší režie, software obsahuje obsluhu ovládacích prvků a jejich zpracování, není tedy třeba dalších komponent pro řízení.

Jiným řešením, které ve své práci využívám, je ovládání pomocí bezdrátového ovladače. Tento ovladač zpracovává řídicí pokyny uživatele, odesílá je na vzdálené zařízení (robot) a ten tyto příkazy zpracovává a interpretuje.

3.1 Robotické vozítko

Robotické vozítko, které ve své práci využívám, jsem si zapůjčil z prostředků fakulty FIT VUT a nese označení SRV-1 Blackfin od firmy Surveyor [1]. Toto vozítko je složeno z motorů pro pohyb vozítka, ovládací desky s konektory, modulem kamery, který je k ovládací desce připojen jedním z konektorů a dvěma lasery. Řídicí deska je připojena k základní desce, připevněné k šasi vozítka, která také obsahuje obvody pro pohon 2 motorů pro pásový podvozek vozítka. SRV-1 je také vybaveno WiFi modulem a anténou pro bezdrátovou komunikaci s jiným zařízením. Celý systém je pak napájen dle specifikace z Li-Pol akumulátorových článků poskytujících napájení 7.2 V 2 Ah [1].

Ve své práci využívám pouze část prostředků, které SRV-1 nabízí - šasi se základní deskou (z důvodu využití obvodu pro řízení motorů) a akumulátoru pro napájení celého modulu. Zbylé součásti budou nahrazeny modulem Arduino FIO s přídatnou elektronikou (např. pro detekci překážek apod.) připojeným k základní desce vozítka. Vozítko s jeho původními komponentami je na Obrázku 3.1.



Obrázek 3.1: Robotické vozítko SRV-1 od společnosti Surveyor [1].

3.2 Ovládací prvky

Ovládacími prvky při řízení vozítek, robotů či jiných zařízení bývají převážně mechanické součástky jako jsou tlačítka, potenciometry, joysticky apod. V závislosti na aplikaci se pak mezi těmito prvky rozhodujeme.

Ve své implementaci jsem zvolil ovládání motorů robota 2 joysticky s integrovaným tlačítkem. Joystick je složen ze 2 potenciometrů (a již zmíněného tlačítka), které jsou vůči sobě otočeny o 90° , čímž sledují vychýlení v osách X a Y.

3.3 Zobrazovací prvky

V závislosti na požadavcích lze využít zobrazovacích prvků jako jsou LED diody, segmentové displeje, řádkové LCD displeje, grafické displeje, případně až monitory či televizory. V této práci jsem pro zobrazení informací o řízení robota využil LCD displej 16x2 znaků s řadičem a podsvícením. Pro signalizaci detekované překážky v okolí vozítka používám 4 LED diody o průměru 5 mm vhodně umístěné na ovladači.

3.4 Komunikační kanál

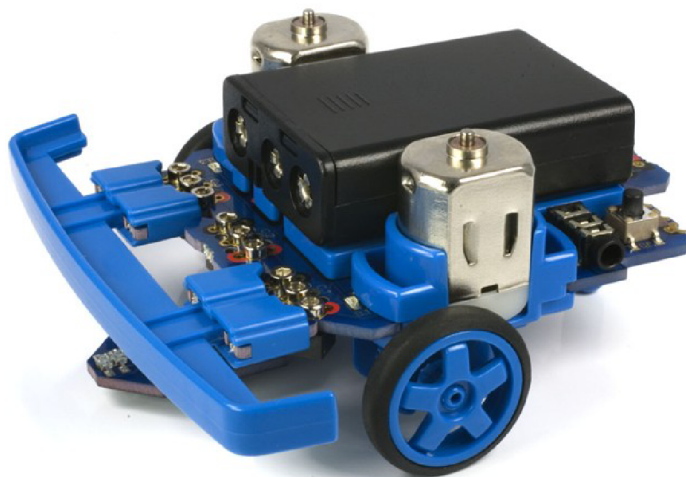
Z hlediska komunikace lze typ spojení rozdělit na přímý (drátový) a bezdrátový. Zvolená platforma Arduino umí komunikovat sériovou linkou přes piny Rx/Tx, protokolem I²C (piny SDA/SCL) a SPI (SS, MOSI, MISO, SCK). Pro bezdrátovou komunikaci nabízí Sparkfun moduly pro protokoly XBee, WiFi, Bluetooth aj. Já jsem zvolil možnost s XBee, jelikož je má fakulta k dispozici a pracují se sériovou linkou desky Arduino.

3.5 Detektory a jiné snímače

Pokud je vyžadováno, aby bylo vozítko/robot schopno autonomní činnosti, případně pro zvýšení bezpečnosti, lze využít elektronických součástí pro detekci elektrických i neelektrických veličin, které podávají požadovanou informaci o stavu vozítka či jeho okolí. Pokud je potřeba kontrolovat překážky, které v pracovním prostoru zařízení mohou existovat, je nutné vozítko vybavit například LED diodou a fototranzistorem pro optické „snímání“ okolí, případně lze využít sonarů, Hallových sond (detekce magnetického pole), mechanických spínačů apod. Pokud jsou požadovány informace o rychlosti, zrychlení, poloze či naklonění robota, je možné použít akcelerometry, GPS moduly, gyroskopy aj.

3.6 Autonomní ovládání

Má-li být ve výsledku robot plně automatizován, tzn. bez zásahu člověka do řízení, neobejde se toto zařízení bez výše zmíněných prvků. Základním autonomním zařízením je pro příklad robot sledující čáru při své jízdě (příklad takového robota od firmy PICAXE je na Obrázku 3.2, případně samovyvažovací robot na jednom kole, který upravuje směr a rychlost motoru kola tak, aby se udržel v rovnovážné poloze. Při složitějších aplikacích je zapotřebí, aby robot ze svých senzorů odečítal informace, tyto informace samostatně zpracoval, vyhodnotil a následně na ně reagoval. Principů posledního zmíněného způsobu bylo využito při aplikaci autonomní činnosti robotického vozítka.



Obrázek 3.2: Robot sledující čáru PICAXE–20X2 Microbot. Obrázek převzat z <http://www.picaxe.com/Hardware/Robot-Kits/PICAXE-20X2-Microbot/>.

Kapitola 4

Návrh a implementace

V praktické části této práce jsou použity níže popsané komponenty, s kterými je dosaženo bezdrátového ovládání robotického vozítka. Při návrhu aplikace bylo zapotřebí stanovit si nejen cíl práce, ale i komponenty, které budou pro dosažení výsledku využity. Postupným pokrokem byly přidávány a testovány jednotlivé periferie modulů spolu s otestováním funkcí dílčích celků. Jako první byly vybrány desky Arduino, které jsou jádrem práce a řídicím prvkem v obou modulech, a to desky UNO a FIO. Od tohoto výběru se pak odvíjela další rozhodnutí, například volba bezdrátových modulů, sonarů apod., s ohledem na to, čím fakulta disponuje.

4.1 Návrh komunikačního protokolu

Při návrhu komunikačního protokolu lze vycházet ze 2 základních variant - lze implementovat takový protokol, kde všechna data budou přenášena pouze jedním směrem a nebude brán zřetel na data z cílového zařízení. Takový protokol se hodí pro typ aplikací, kde je potřeba řídit jedno zařízení druhým bez zjišťování stavu druhého zařízení, nebo pokud je potřeba z jednoho zařízení data posílat jinam bez potřeby na ně odpovídat.

Druhou možností je komunikační protokol postavit na vzájemné komunikaci, kde jedno zařízení vyšle data do druhého a to na ně reaguje příslušnou odpovědí. Odpověď se může a nemusí týkat právě obdržených dat - odpověď může být závislá na přijatých datech například pro bezdrátové kódové zámky (příklad: uživatel zadá kód, ten je odeslán do řídicí jednotky a ta odpoví příkazy odemknout / nechat uzamčené). Naopak aplikace nezávislé na přijaté zprávě (odpovědi se tato data netýkají a ani je nijak neovlivňují) mohou být například různá řízení robotů, kde jsou na staně ovladače odeslána data na ovládání cíle, a tento cíl pak odpoví zprávou o svém stavu, poloze atp. Tato data spolu nijak nesouvisí, ale přesto je potřeba zajistit obousměrný komunikační provoz, což může být v některých aplikacích, náročných na okamžité reakce, problém.

Je také potřeba navrhnout způsob odesílání a přijetí přenášených dat. Pokud již máme vybraný komunikační standard, můžeme začít implementovat komunikační protokol. Existují mnohé způsoby, nicméně se zde pokusím popsat pouze několik základních variant s ohledem na směr přenosu, typ přenášených dat a také na frekvenci přenosů.

Přenášená data mohou nabývat různých datových typů. Pokud v naší aplikaci využijeme pouze jediného, je vhodné zvolit mezi jednotlivými posloupnostmi dat **oddělovač**. Ten nám oddělí jeden údaj od druhého, což nám zajistí, že na straně příjemce budeme vždy přesně vědět, kdy začíná nová hodnota. Může se jednat o jediný znak, či o sekvenci

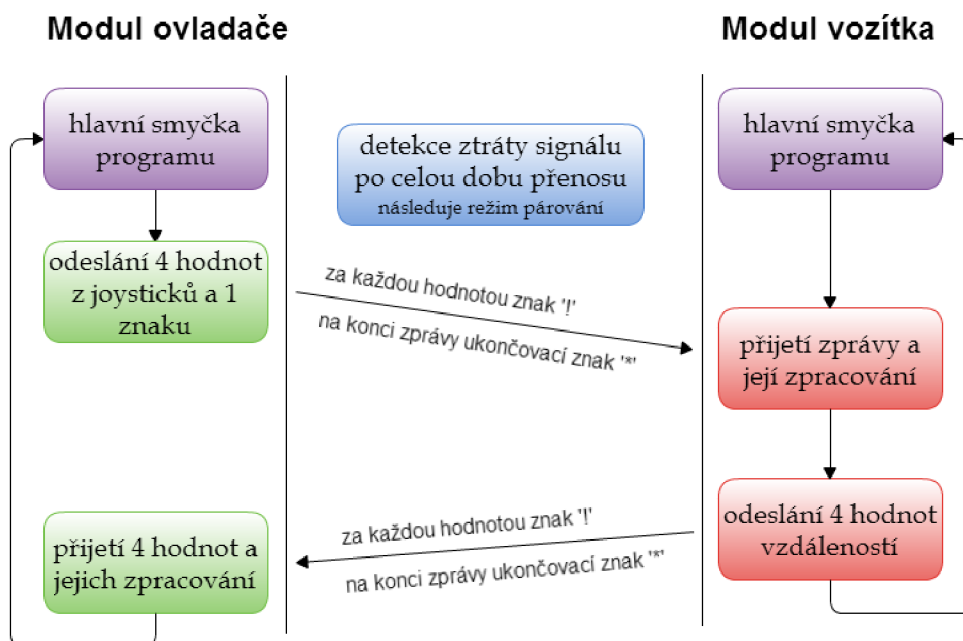
znaků (např. řetězec "STOP"). Také je potřeba zvolit oddělovač celých zpráv (zejména pro jednosměrnou komunikaci), aby bylo jasné, kdy končí přenos minulé zprávy a začíná přenos první hodnoty nové zprávy. Tento oddělovač je důležitý zejména v případě, pokud přenášíme data ve smyčce s určitým pořadím, potřebujeme vědět, kdy přenos začne, abychom mohli synchronizovat stranu příjemce se stranou odesílatele. Může se totiž stát, že se spojení dvou bezdrátových zařízení naváže až v půlce odesílané zprávy (případně v jiné části zprávy), tudíž se všechna data do začátku dalšího cyklu zahodí a přijímač vyčká na synchronizační oddělovač počátku zprávy. Pak díky oddělovačům jednotlivých hodnot přijímač ví, kde hodnota dané informace začíná i končí.

Podle datového typu odesílaných dat můžeme zvolit typ oddělovače zpráv i samotných dat. Je vhodné zvolit oddělovač jiného datového typu než odesílaná data, nicméně (např. v systémech, kde je využito více datových typů přenášených dat) je volba oddělovače obtížná, proto je plně na programátorovi, jaký zvolí.

4.2 Zvolené řešení

V mé implementaci jsem nejdříve zvolil komunikační protokol (s ohledem na zasílaná data) jednosměrný a v opakující se nekonečné smyčce. Z ovladače jsou odesílána řídicí data do robotického vozítka, které musí okamžitě reagovat na změny polohy ovládacích prvků na ovladači. Z tohoto důvodu nelze uvažovat o přerušení spojení (uspání a vyčkání na další přenos za určitý časový úsek). Tyto hodnoty jsou číselné, mezi nimi jsem zvolil oddělovačí znak '*', který cílové zařízení rozpozná, uloží hodnotu a načte novou. Po konci smyčky následuje znak '!', po kterém obě strany přejdou k nové iteraci přenosu.

Jako rozšíření byl implementován obousměrný přenos, kde vozítko odesílá zpět ovladači údaje o svém stavu (detekce překážek) a tento stav bude zobrazován uživateli na displeji modulu ovladače. Dalším přídatkem bylo umožnění přechodu do autonomního režimu (více v kapitole 4.7.6). Diagram obousměrného přenosu je znázorněn na Obrázku 4.1.



Obrázek 4.1: Diagram komunikace modulů při zasílání zpráv.

4.3 Návrh modulů

Z pohledu návrhu modulů je práce rozdělena na 2 základní části - modul ovladače a modul vozítka. Oba moduly jsou osazeny deskou Arduino s připojenými podpůrnými periferiemi a obvody, které zajišťují požadované funkce definované v návrhu. V souhrnu jsou to tyto požadavky:

- využití platformy Arduino,
- ovládaní robotického vozítka bezdrátově,
- zobrazování informací o pohybu vozítka či jiných statistik,
- detekce překážek v okolí vozítka,
- maximální využití V/V vývodů desek Arduino,
- rampa pro světlo či kameru.

4.4 Použité periferie

V Tabulce 4.1 jsou sepsány komponenty, ze kterých byly výsledné moduly poskládány. Rozšiřující desky, spolu se senzory, řídicími prvky a zobrazovači, byly vyrobeny / zakoupeny, vozítko, desky Arduino a komponenty pro bezdrátovou komunikaci jsou zapůjčeny ze školních prostředků. Periferie byly vybrány s ohledem na vhodné použití s platformou Arduino (dostupnost knihoven pro ovládání).

	Modul ovladače	Modul vozítka
Řídicí deska	Arduino UNO	Arduino FIO
Napájení	9 V akumulátor	7.2 V akumulátor
Rozšíření	XBee shield, vlastní deska pro UNO	XBee shield, vlastní deska pro FIO
	LCD displej, 4 LED 2 joysticky vypínač napájení modulu vypínač podsvícení displeje	světelná rampa se 3 LED ¹ 2 modelářské serva 4 ultrazvukové sonary

Tabulka 4.1: Použité periferie u obou modulů.

4.4.1 Zobrazovače

Pro zpětnou vazbu uživatele s vozítkem byl na modul ovladače umístěn LCD displej GDM-1602A, obsahující 16x2 zobrazovacích znaků, na kterém se uživateli zobrazují informace o vytížení levého a pravého motoru v procentech, s možností přepnutí režimu na zobrazování vzdáleností jednotlivých stran vozítka od překážek. Tento displej také ukazuje režim párování, pokud dojde k jeho aktivaci při komunikaci obou modulů. Lze také ovládat jeho podsvícení, kterým disponuje, a to mechanickým přepínačem na ovladači.

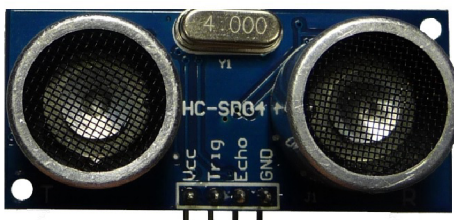
¹Světelná rampa je popsána v 4.4.5

4.4.2 Řídicí prvky

K ovládní slouží dvojice joysticků, pohyblivých ve 2 osách a integrující tlačítko, které je aktivní při stisku kloboučku joysticku. Díky tomuto páru joysticků je možné ovládat zároveň pohyb robota i světelné rampy umístěné na robotu, spolu s naprogramovanými funkcemi po stisku kloboučků (viz 4.7.4).

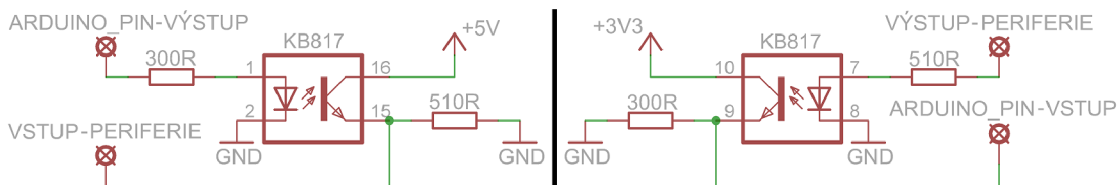
4.4.3 Detektory a snímače

Po stranách robota byly umístěny 4 ultrazvukové senzory HC-SR04 (Obrázek 4.2), aby mohl robot poskytovat údaje o nebezpečí nárazu v jeho blízkém okolí. Primárně tyto senzory slouží k měření vzdálenosti pomocí ultrazvukového impulsu a příjmu jeho odrazu, z čehož pak program (modul ovladače) tyto hodnoty po přijetí zpracovává a vyhodnocuje. Tento senzor podle technické specifikace [4] dokáže měřit vzdálenosti v rozmezí 2–400 cm s přesností 0.3 cm. Pro účely práce je tato maximální hodnota programově omezena na vzdálenost 100 cm.



Obrázek 4.2: Ultrazvukový senzor vzdálenosti HC-SR04.

Jelikož však senzor pracuje s napětím 5 V, zatímco použité Arduino FIO operuje s 3.3 V, bylo potřeba vytvořit převod napěťových úrovní pro tyto senzory, ale také pro serva, které rovněž pracují s 5 V. Tohoto jsem docílil pomocí integrovaného obvodu KB847 [9] se 4 optočleny, u nichž (v závislosti na funkci vývodu dané periferie) je na vstupu (výstupu) přivedena úroveň napětí 3.3 V či 5 V a opačná úroveň na výstupu. Pokud má tedy určitý vývod funkci výstupu (z pohledu Arduino), je přes odpor (pro omezení proudu vstupem optočlenu) přivedeno napětí 3.3 V (logická úroveň 1), na výstupu optočlenu je pak napětí 5 V. Schématické zapojení je na Obrázku 4.3.

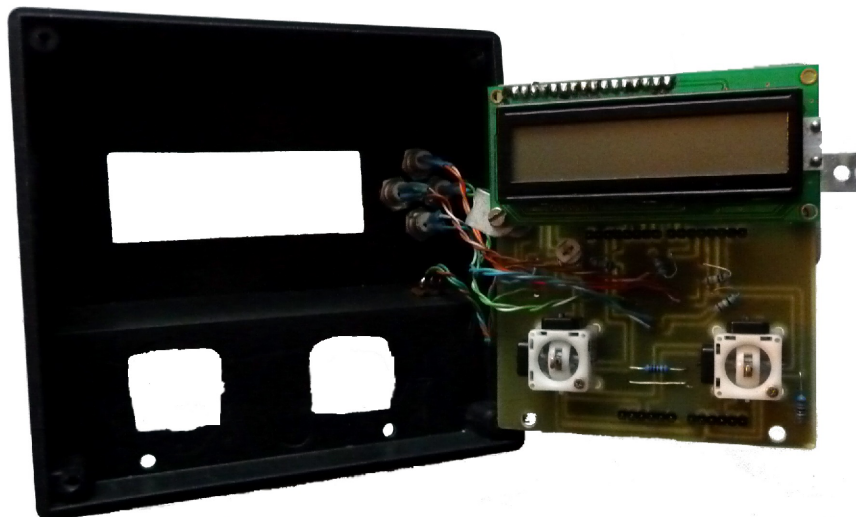


Obrázek 4.3: Schématické zapojení s použitím optočlenu KB847. Vlevo zapojení pro ovládní periferieho zařízení (výstupní vývod Arduino), vpravo pak zapojení pro čtení hodnoty ze zařízení (vstupní vývod).

4.4.4 Konstrukční prvky

Pro připevnění světelné rampy na šasi vozítka bylo využito stavebnice **Merkur**, která svou univerzálností a snadným spojováním umožňuje široké využití i v oblasti robotiky. Modul

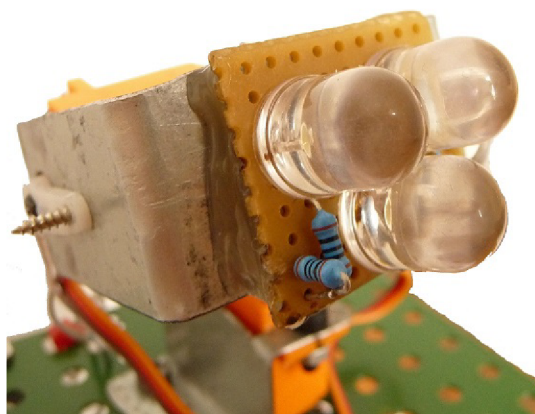
ovladače, kromě samotného plastového boxu, ve kterém jsou všechny součásti ovladače umístěny (Obrázek 4.4), obsahuje 4 distanční sloupky 22 mm pro uchycení LCD displeje k rozšiřující desce modulu ovladače. Ke spodní straně DPS je tekutým lepidlem připevněna patice pro 9V baterii.



Obrázek 4.4: Umístění modulu ovladače do konstrukční krabičky U-KP18.

4.4.5 Světelná rampa

Světelná rampa, kterou pohybují 2 servomotory, osvětluje prostředí kolem robotického vozítka pomocí trojice supersvítivých LED diod, konkrétně typ OSW5DKA131A [11]. Jedná se o 10 mm LED diodu se svítivostí 30 000 mcd a proudem v propustném směru 30 mA. LED diody jsou napájeny na univerzálním plošném spoji spolu s odpory v sérii, všechny 3 větve jsou v paralelním zapojení. Z tohoto plošného spoje pak vedou 2 přívodní vodiče, kterými je řízen svit rampy. Detail tohoto modulu zachycuje snímek na Obrázku 4.5.



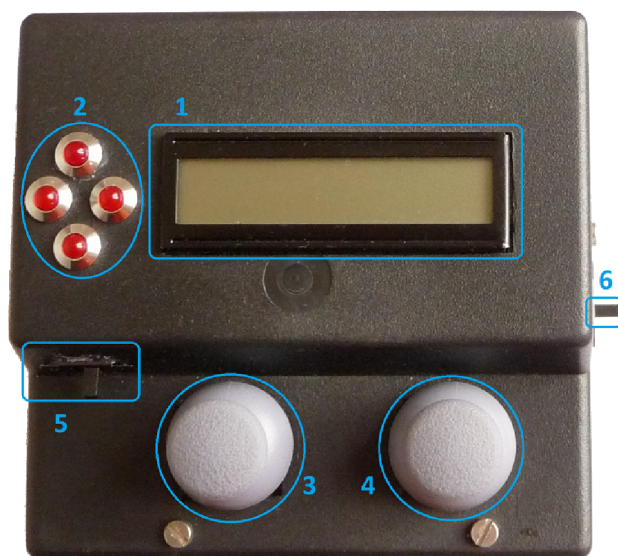
Obrázek 4.5: Detail světelné rampy robotického vozítka.

4.4.6 Motory

Hybnost celého robota zajišťuje dvojice motorů umístěných v šasi podvozku robotického vozítka SRV-1, popsaného v kapitole 3.1. Pro pohyb světelné rampy byla využita dvojice modelářských servomotorů HK15178, které pohybuji s rampou ve dvou osách. Náklon obou os a tedy ovládání servomotorů zajišťuje pravý z dvojice joysticků umístěných na ovladači. Stisk tlačítka tohoto joysticku ovládá úroveň svitu světelné rampy.

4.5 Modul ovladače

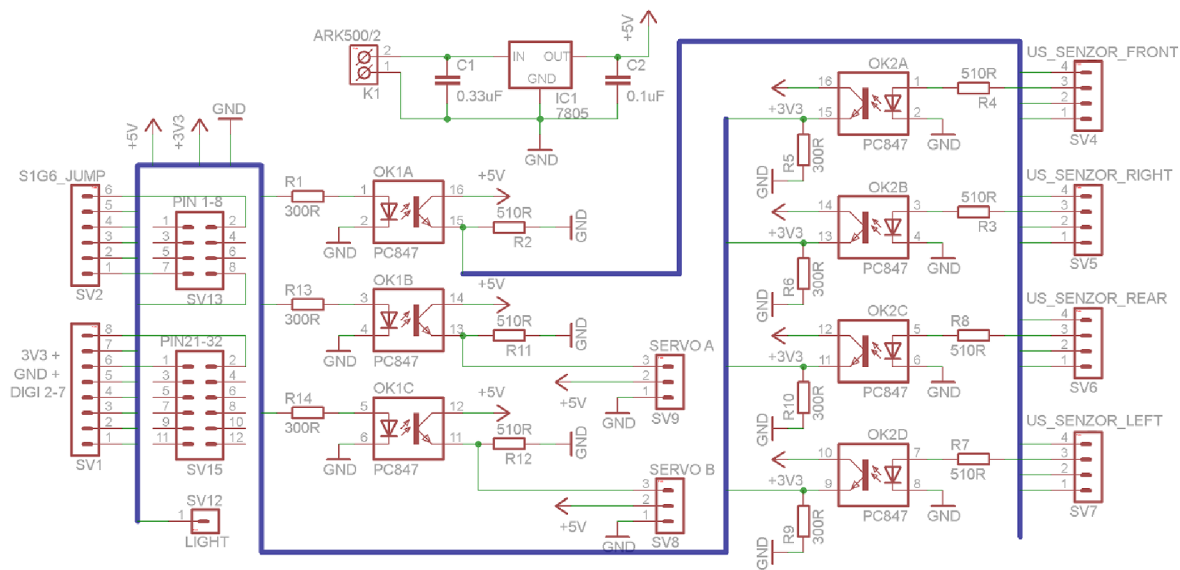
Modul ovladače je ovládacím nástrojem pro pohyb robotického vozítka, nenachází-li se v autonomním režimu (kapitola 4.7.6). Na Obrázku 4.6 je tento modul vyobrazen spolu s označením jednotlivých komponent, jež jsou uloženy v plastovém konstrukčním boxu. Funkce těchto komponent jsou sepsány v Tabulce 4.2.



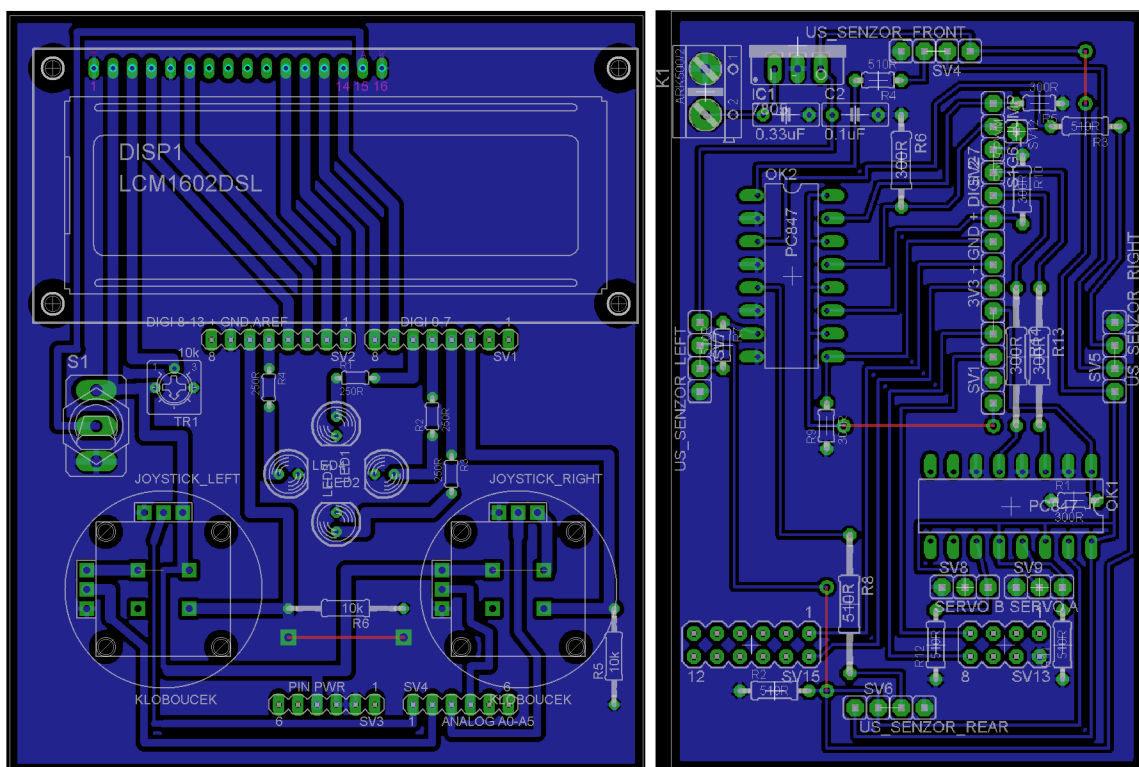
Obrázek 4.6: Modul ovladače - čelní pohled. Čísla komponent spolu s jejich funkcí jsou uvedena v Tabulce 4.2.

Pro rozšiřující shield bylo vytvořeno schéma zapojení (Obrázek 4.7) i návrh desky plošných spojů (Obrázek 4.9 - vlevo). Tato deska byla vyrobena ve školních laboratořích, osazena součástkami a připojena k desce Arduino UNO přes XBee shield. Po osazení byla sestava doplněna o externí komponenty (LED diody, napájení spolu s držákem 9 V baterie a mechanické vypínače) a připevněna k plastové krabičce. Krabička je složena ze dvou částí, dna (víka) a horní části, ke které je modul připevněn, aby bylo možné spodní kryt jednoduše oddělit od krabičky. Z toho důvodu byl také držák na baterii přilepen tavnou pistolí k rozšiřující desce modulu ovladače.

na Obrázku 4.9 (vpravo).



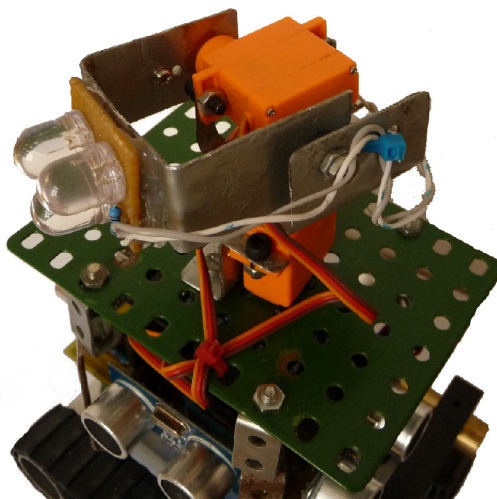
Obrázek 4.8: Schématické zapojení rozšiřující desky pro Arduino FIO.



Obrázek 4.9: Schématické zapojení rozšiřujících desek pro modul ovladače (Arduino UNO – vlevo) a modul vozítka (Arduino FIO – vpravo).

Světelná rampa byla připevněna k šasi vozítka pomocí dílů stavebnice Merkur s mož-

ností její případné snadné demontáže. Na horním dílu konstrukce, který zároveň tvoří horní kryt celého robotického vozítka, je připevněna světelná rampa v podobě upevnění 2 modelářských serv v osách pojmenovaných A a B vzájemně otočených o 90°. Servo v ose A otáčí s rampou kolem osy Y kartézského souřadného systému, druhé servo v ose B pak kolem osy Z. Na rameni druhého serva je pak tekutým lepidlem přimontován univerzální plošný spoj se třemi LED diodami, které tvoří vlastní světelnou rampu vozítka. Detail této konstrukce je na Obrázku 4.10.



Obrázek 4.10: Konstrukce světelné rampy robotického vozítka.

4.7 Řídicí program

Při tvorbě řídicího programu obou modulů byl zvolen iterativní přístup, kdy bylo postupováno od jednoduchých operací ke složitějším. Nejdříve bylo potřeba naprogramovat XBee moduly tak, aby vytvořily vzájemné spojení a znaly své adresy. K této operaci slouží modul XBee Explorer znázorněný na Obrázku 2.6. Při nastavování parametrů modulů byl také znemožněn přechod modulu XBee do režimu spánku mezi intervaly komunikace (přenosu) kvůli zaručení okamžité reakce na pokyny z ovladače.

Při vytváření zdrojových kódů bylo využito zmiňovaného vývojového prostředí Arduino IDE, případně textového editoru PSPad. Arduino IDE umožňuje naprogramovat vytvořený skeč přímo do vybraného zařízení Arduino, nicméně u modulu ovladače bylo nutné odpojit bezdrátový modul XBee, případně přepínačem na rozšiřujícím XBee shieldu přepnout do režimu USB, jelikož XBee i USB využívají vývody sériové linky Arduino. Stejný problém nastává i u modulu vozítka, nicméně ten postrádá přepínač pro možnost změny režimu a to z důvodu přímé integrace patice pro XBee modul na desku Arduino FIO (nenachází se zde XBee shield) a také z nutnosti využít speciální modul pro naprogramování desky FIO přes USB.

4.7.1 Použité knihovny

Pro ovládání LCD displeje, servomotorů a ultrazvukových senzorů bylo zapotřebí importovat příslušné knihovny pro ovládání těchto periférií pomocí knihovnických funkcí. U obou modulů byla vytvořena samostatná knihovna s potřebnými konstantami, proměnnými a makry

pro názvy a čísla vývodů pro daný modul. Knihovny jsou importovány za úvodními řádky zdrojových kódů, na nichž jsou iniciály autora spolu se stručnými informacemi o kódu.

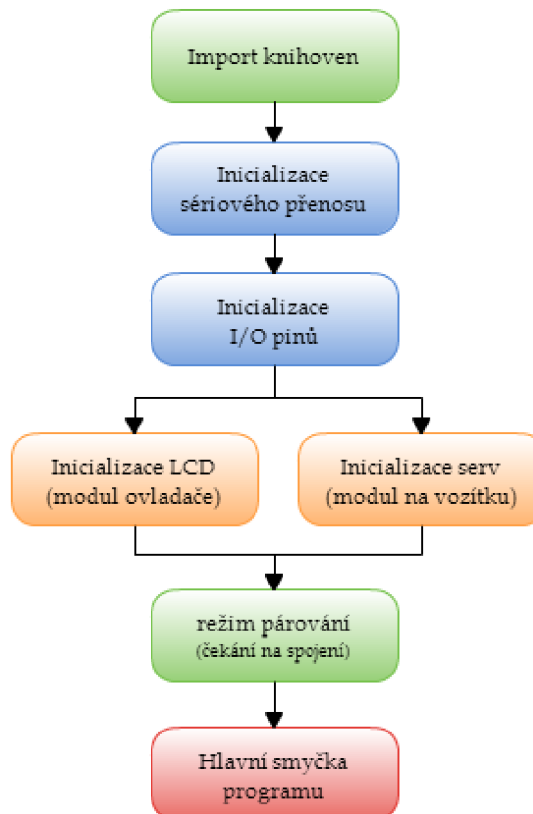
Knihovny pro zmíněné periferie byly použity z knihoven integrovaných ve vývojovém prostředí Arduino IDE, kromě knihovny pro ultrazvukové snímače - tato knihovna byla převzata a upravena ze stránek <http://blog.iteadstudio.com/arduino-library-for-ultrasonic-ranging-module-hc-sr04/>, jak je zmíněno ve zdrojovém kódu samotné knihovny.

4.7.2 Inicializace

V inicializační funkci `setup()` probíhá inicializace všech vstupně/výstupních vývodů obou zařízení (definice jejich funkce - je-li vstupní či výstupní), spolu s inicializací příslušných periferních zařízení (LCD displej apod.). Na Obrázku 4.11 je znázorněna pomocí pseudokódu inicializace obou modulů.

U modulu ovladače dochází postupně k aktivaci sériové linky, inicializaci V/V vývodů, inicializaci LCD displeje, vypsání uvítací zprávy na displeji a nastavení modulu do režimu párování (o němž informuje zpráva na displeji a rozsvícená horní ze čtveřice signalizačních LED diod). U inicializace LCD je nejprve vytvořen objekt, jehož parametry jsou použité řídicí vstupní a výstupní vývody Arduino (v předem definovaném pořadí). Poté je ve funkci `initLCD()` definována velikost LCD displeje (v tomto případě 16 znaků a 2 řádky), načež jsou definovány požadované vlastnosti pro kurzor a výpis na LCD.

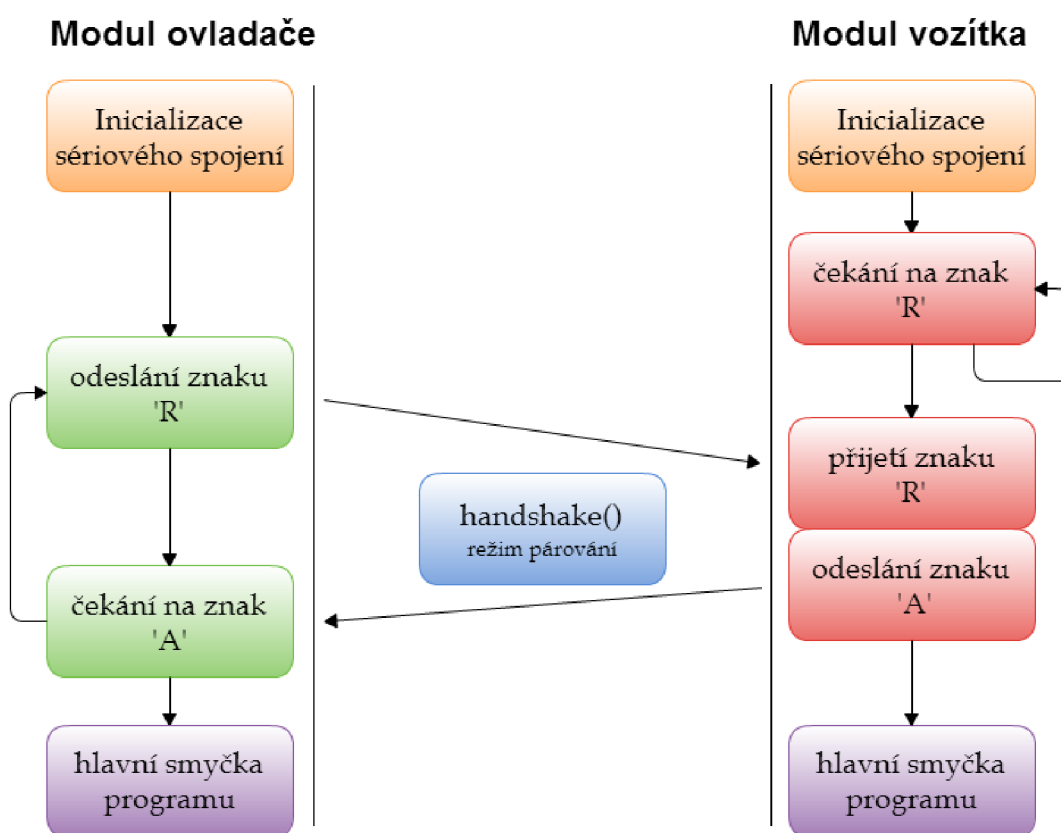
U modulu vozítka probíhá inicializace podobně, avšak namísto inicializace LCD displeje jsou vytvořeny objekty pro oba servomotry a uvedení jejich natočení do polohy 90°. Následně je modul uveden taktéž do režimu párování, čímž inicializační část končí.



Obrázek 4.11: Diagram inicializační části programu obou modulů.

4.7.3 Režim párování

Po úvodní inicializaci obou modulů může započít i úvodní komunikace mezi moduly. Funkce **handshake()** zajišťuje spárování a ustavení komunikace tak, aby oba moduly věděly, že mohou začít komunikovat, nebo v případě výpadku signálu vyčkat na obnovení spojení. U modulu ovladače je v této funkci odeslán znak 'R' pro žádost o párování, je-li sériová linka volná. Pokud po prodlevě 10 ms nepřišel znak 'A', značíci potvrzení žádosti o párování, je tento postup vykonáván v nekonečné smyčce a zařízení tak pouze sleduje stav sériové linky. Po úspěšném spárování je u modulu ovladače vypsána zpráva o úspěšném spárování a program následně pokračuje hlavní smyčkou **loop()**. Analogicky pak párování probíhá u modulu vozítka, kdy je očekáván znak 'R'. Odpovědí na jeho přijetí je odeslání znaku 'A' a program poté taktéž začne vykonávat hlavní obslužnou rutinu. Grafické znázornění této komunikace je na Obrázku 4.12.



Obrázek 4.12: Diagram průběhu komunikace při párování modulů.

Jak již bylo naznačeno, režim párování je aktivován nejen po zapnutí modulu během jeho inicializace, ale i v chybových situacích, jako je ztráta signálu (spojení) mezi moduly, případně při příliš dlouhé odezvě na zaslouanou zprávu. Limitní čas pro aktivaci režimu párování byl stanoven na 2000 μs u obou zařízení. U modulu ovladače je opět uživateli pomocí LCD displeje tato událost signalizována výpisem hlášení a signalizační LED diodou. Pro zajištění bezpečnosti vozítka v případě výpadku signálu je ve funkci pro párování přidán příkaz pro zastavení vozítka.

4.7.4 Režim přenosu

V hlavní smyčce programu se moduly nacházejí ve dvou režimech - režim přenosu a režim příjmu. O tomto rozpoložení rozhodují proměnné dvouhodnotového datového typu, které se v každé fázi překlopí na opačnou hodnotu (při režimu přenosu se nastaví na režim příjmu, aby v další iteraci hlavní smyčky programu tento příjem proběhl).

Modul ovladače

Modul ovladače po spárování zahájí přenos informací druhému modulu, formát této komunikace je popsán v sekci 4.2. Nejdříve je ověřeno, zda příslušné příznaky značí režim přenosu a to, že se zařízení nenachází v autonomním režimu. Poté jsou přečteny hodnoty z analogových vstupů (4 hodnoty joysticků), které jsou mapovány z užitečných hodnot (vlivem umístění v plastové krabičce a vlivem opotřebení potenciometrů joysticků) na hodnoty z intervalu $-255 - 255$ (levý joystick), resp. na hodnoty $-30 - 30$ (pravý joystick). V rámci zamezení pohybu v centrální pozici kloboučku joysticku, případně jemného dotyku kloboučku, je vyfiltrována tzv. *mrtvá zóna*, která tyto drobné odchylky od středu (hodnoty 0, 0) eliminuje. Taktéž v případě, že by se namapovaná hodnota vyskytla mimo interval $-255 - 255$, resp. $-30 - 30$, je tato horní či dolní mez ořezána na krajní hodnoty.

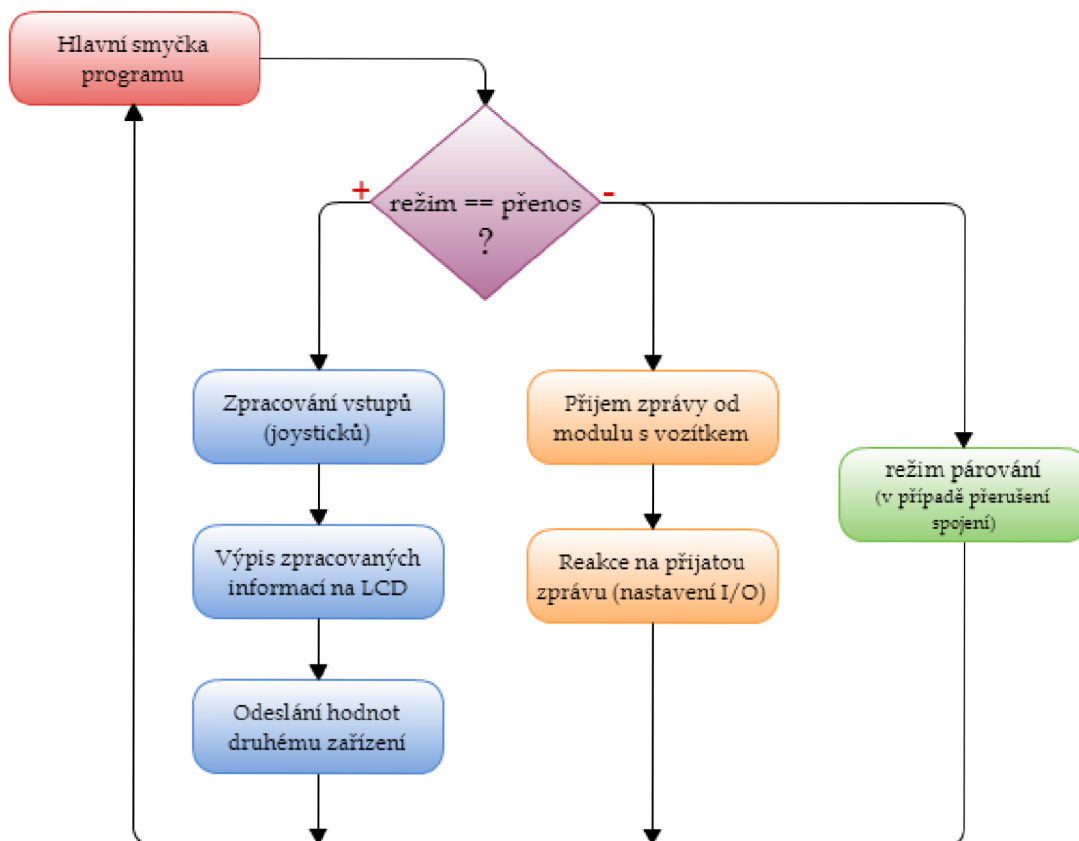
Jelikož bylo u modulu vozítka pro řízení pohybu použito ovládání 1 joystickem levý a pravý motor pásového podvozku, algoritmus pro převod na tento způsob řízení byl inspirován z webových stránek². Poté jsou tyto vypočtené hodnoty převedeny na procenta, kvůli následnému režimu vypisování procentuálního vytížení levého a pravého motoru. Následuje už samotný přenos těchto 4 hodnot (údaj pro levý motor, údaj pro pravý motor, úhel otočení servomotoru A, úhel otočení servomotoru B) přes sériovou linku spolu s jejich oddělovači (jak je znázorněno na diagramu na Obrázku 4.1).

Pro oba režimy jsou implementovány algoritmy pro převod z číselných hodnot na formátovanou reprezentaci těchto hodnot na LCD displeji. Z důvodu zpomalení mazání a nového výpisu informací na LCD, bylo implementováno časové omezení výpisu v intervalech po 100 ms. Diagram programu modulu ovladače je na Obrázku 4.13.

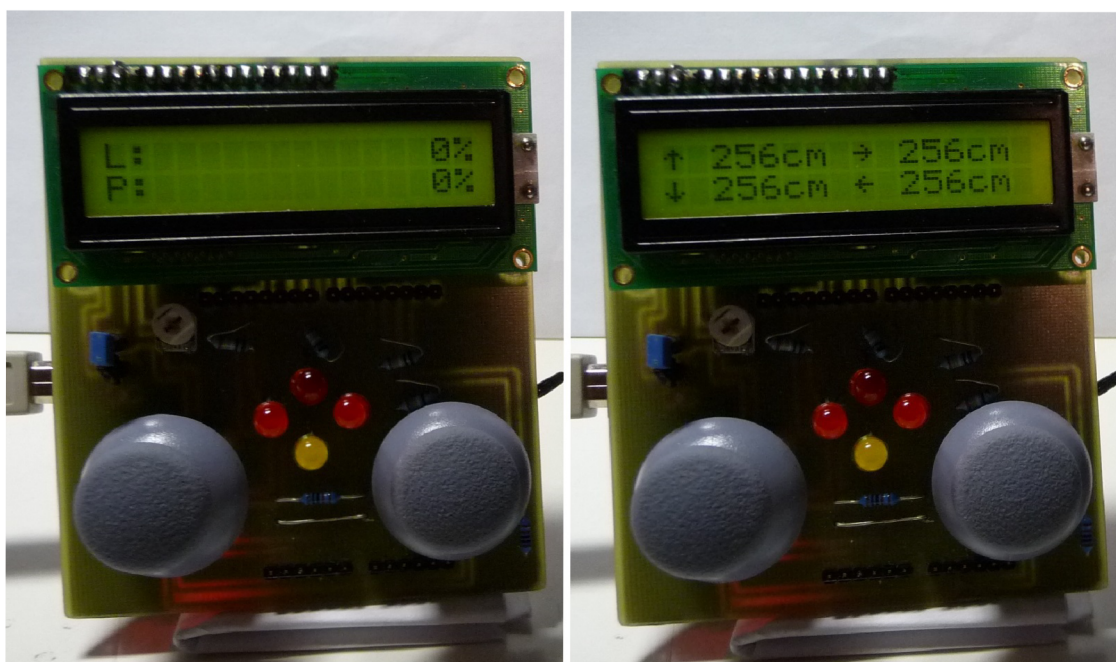
Kromě analogových hodnot je z joysticků odečítán i digitální signál z integrovaného tlačítka pod kloboučkem samotného joysticku. Tato tlačítka byla použita pro implementaci rozšířených funkcí pro ovládání vozítka. Tlačítka na levém joysticku slouží pro přepínání zobrazovacího režimu LCD displeje, a to procentuálního vytížení motorů, či zobrazování vzdáleností překážek od vozítka (Obrázek 4.14). Tlačítkem na pravém joysticku je regulován svit světelné rampy robotického vozítka. Jsou-li stisknuty obě tlačítka, aktivuje se režim autonomního řízení vozítka, který je podrobně popsán v sekci 4.7.6.

V této fázi programu tedy dojde k přečtení stavů tlačítek joysticků, kdy se nejprve uloží pravdivostní hodnoty značící jejich stisknutí, načež pak dojde k rozhodnutí, která speciální funkce bude aktivována podle výše zmíněného pravidla. Pokud je aktivován autonomní režim, odešle modul ovladače znak 'A', v případě stisku pouze pravého joysticku je odeslán znak 'H'. V ostatních případech je odeslán druhé straně znak 'L'. Následně je komunikace ukončena zasláním ukončovacího znaku '**', nastaveny příslušné příznaky a v následujícím opakování hlavní smyčky programu bude modul ovladače vyčkávat na data, která zašle modul vozítka.

²Originální algoritmus byl zveřejněn v příspěvku na webových stránkách <http://electronics.stackexchange.com/questions/19669/algorithm-for-mixing-2-axis-analog-input-to-control-a-differential-motor-drive/19702#19702>, zveřejněn uživatelem Kamil Zadora. Citováno dne 11.02.2013.



Obrázek 4.13: Diagram programu modulu ovladače.

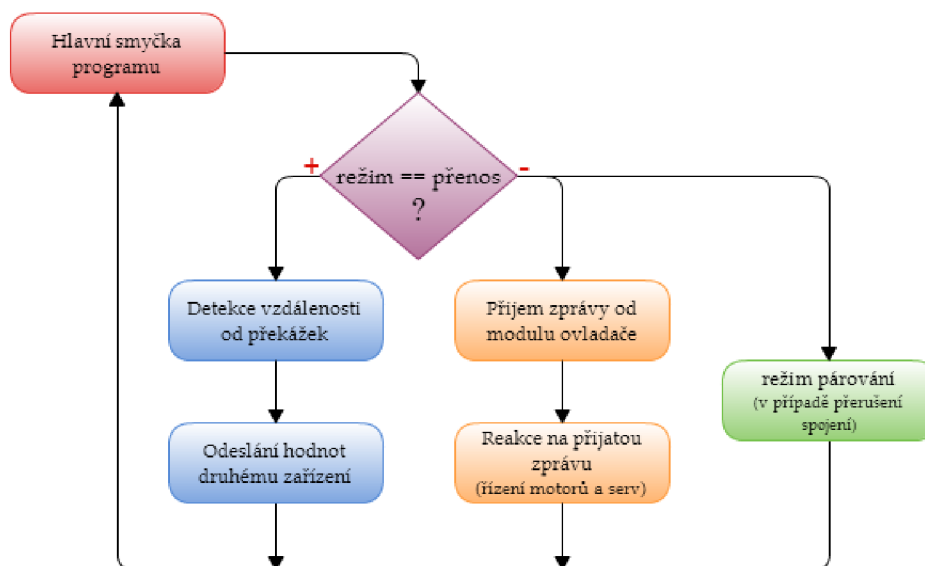


Obrázek 4.14: Ukázka zobrazovacích režimů LCD displeje u modulu ovladače.

Modul vozítka

Modul vozítka má při přenosu pouze jeden úkol - změřit vzdálenosti od překážek v jeho okolí pomocí ultrazukových senzorů a odeslat je modulu ovladače. Měření probíhá podle postupů uvedených v [4] - tedy změnu úrovně na vysílacím vývodu modulu po dobu $10\ \mu\text{s}$ a na přijímacím vývodu změřit čas od odeslání do změny na tomto vývodu na vysokou úroveň. Tato operace, včetně převedení času na vzdálenost byla implementována v knihovně pro ultrazukový modul (zdrojový kód v Příloze B).

Jelikož však ve snaze co nejvíce urychlit přenos a zároveň nutnosti měřit vzdálenosti ze všech 4 stran docházelo při měření k *preslechům*³, bylo nutné pro zmírnění tohoto jevu měření opakovat a upravit pořadí měření jednotlivých stran - bylo využito vícenásobné měření ve stejném okamžiku, a výsledné hodnoty zprůměrovány tak, aby zkeslení bylo co nejmenší. Po změření všech 4 stran jsou tato data odeslána modulu ovladače, načez je nastaven příznak na režim příjmu a iterace cyklu končí. Pseudokód pro hlavní smyčku programu je znázorněn na Obrázku 4.15.



Obrázek 4.15: Diagram programu modulu vozítka.

4.7.5 Režim příjmu

V režimu příjmu modul ovladače očekává zprávu se 4 hodnotami od modulu ovladače, zatímco modul vozítka očekává řídicí pokyny v podobě 4 hodnot pro motory a jednoho znaku.

Modul ovladače

Při příjmu modul vyčká na data na sériové lince, které pak následně znak po znaku čte a zpracovává. V případě ukončovacího znaku opustí režim čtení, nastaví příznak režimu na

³Tento jev byl pravděpodobně způsoben opožděnými odrazy signálu z jiného senzoru, než který právě měří - výsledné chování pak vykazuje náhodné či nepřesné hodnoty vzdálenosti při neměnných vzdálenostech od překážek.

režim přenosu a přijaté informace uloží do patřičných proměnných, přičemž je také vyhodnoceno, zda-li se hodnoty vzdálenosti od překážek nacházejí v limitní vzdálenosti od vozítka. Podle těchto vzdáleností se pak rozhodne, zda-li bude aktivována signalizace některé ze 4 LED diod, informující o blízké překážce uživatele na modulu ovladače. V případě přijetí oddělovače mezi hodnotami dočasnou načtenou hodnotu (číselnou) uloží do pole hodnot a pokračuje čtením znaků. V ostatních případech jsou data zpracována jako čísla spolu se znaménkem pro záporná čísla.

Modul vozítka

Podobně jako u modulu ovladače je o prováděné operaci rozhodováno přijatým znakem. V případě ukončovacího znaku přenosu jsou přijaté hodnoty pomocí speciální funkce vloženy na výstupy a tudíž i na příslušné periferie. K natáčení servomotorů jsou použity knihovní funkce, k ovládání motorů vozítka je využita DPS s H-můstkem. Toto řízení vyžaduje užití 2 V/V vývodů - jeden pro definici směru (L pro směr vzad, H pro směr opačný) a druhý pro rychlost otáčení motoru (zde je využito PWM modulace pro zajištění plynulého pohybu). Pokud se vozítka nenachází v autonomním režimu, jsou pro pohyb vozítka použity hodnoty odeslané ovladačem, v opačném případě pak autonomní režim sám zvolí směr pohybu (rychlost je pevně stanovena). Dále jsou nastaveny hodnoty úhlu natočení obou servomotorů s omezením na interval $0^\circ - 180^\circ$, jinak by došlo k poškození motorů. Poté jsou opět nastaveny příznaky pro režim přenosu a příjem ukončen. Při přijetí oddělovače je analogický postup jako již výše zmíněný u modulu ovladače. Odlišnost je zde pouze v přijetí jednoho znaku, který symbolizuje stav tlačítek joysticků na modulu ovladače. Tento je pak zpracován a aktivován autonomní režim, nebo zvýšena úroveň svitu LED světelné rampy o 1/3 maximálního svitu (v případě plného svitu je další stisk tlačítka považován za zhasnutí všech LED).

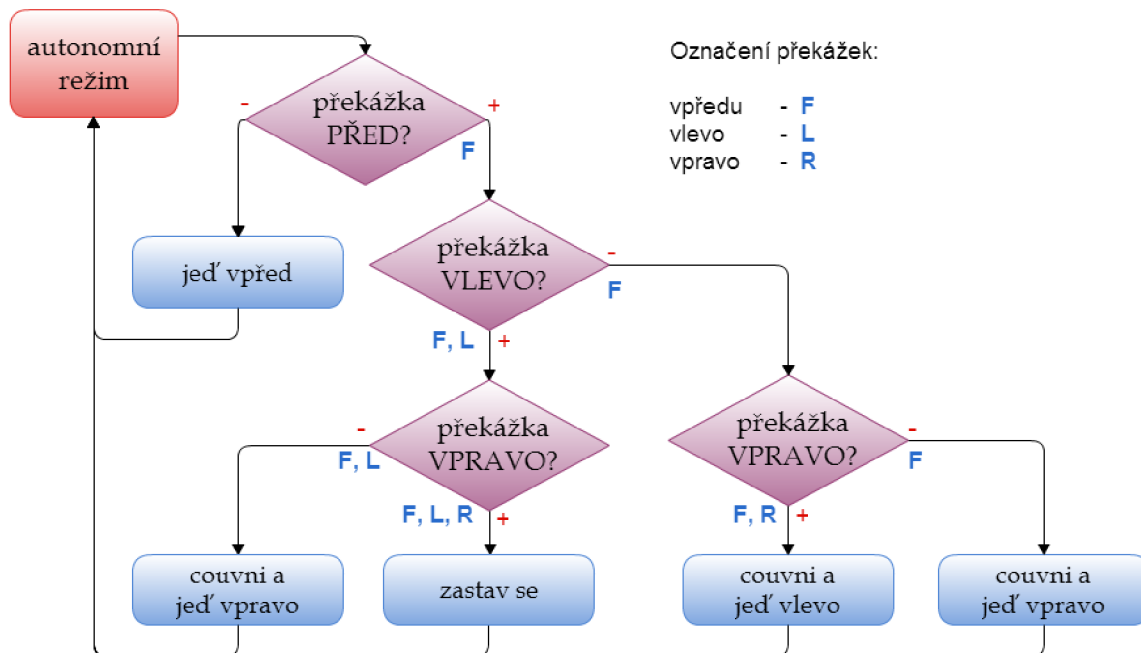
4.7.6 Autonomní režim

Na Obrázku 4.16 je znázorněn pseudokód s rozhodovacím algoritmem pro řízení pohybu robota. Díky ultrazvukovým sensorům robot sám rozpozná překážku v daném směru a podle zmíněného algoritmu určí nový směr pohybu, případně pokračuje dále rovně či zastaví. Po testování autonomního režimu byla zvolena minimální vzdálenost od překážky na 30 cm. Při detekování překážky (nacházející se ve vzdálenosti kratší než zmíněných 30 cm) ve směru jízdy, vozítko zjistí, je-li volno na levé straně. Nezávisle na výsledku vozítko otestuje i překážku vpravo, přičemž reakce se liší. Pokud jsou detekovány překážky ze všech 3 stran vozítka, vozítko zůstane stát na místě dokud tato situace nepomine (v případě pohyblivých překážek). Je-li překážka pouze na jedné z bočních stran, vozítko krátce couvne a otočí se na příslušnou stranu. Krátké couvnutí bylo implementováno kvůli setrvačnosti vozítka a zpoždění v měření - vozítko překážku detekuje, nicméně než dojde k jeho zastavení či změně směru, urazí ještě určitou dráhu. Proto je jako kompenzace této setrvačnosti při detekované překážce na jedné ze stran vozítka vložen pohyb vzad a až poté úprava směru. Nenachází-li se překážka jinde než ve směru jízdy, vozítko se otočí vpravo.

Detekce překážky za vozítkem je sice aktivní, v autonomním režimu je však zanedbána a není proto využita pro rozhodování o směru jízdy.

Při autonomním režimu není možné vozítko ovládat joysticky - tzn. pohyb vozítka ani pohyb rampy, zachovány jsou však funkce tlačítek joysticků. Lze tedy vozítko opět přepnout zpět do manuálního režimu, lze přepínat zobrazovací režim na LCD displeji ovladače a také

měnit osvětlení světelné rampy. To je důsledek úpravy komunikačního protokolu - v autonomním režimu se odesílá namísto 4 hodnot z joysticků (včetně jejich oddělovačů) a 1 znaku, který symbolizuje stav tlačítka pravého joysticku - nestisknuto ('L'), stisknutý pravý joystick ('H'), stisknuty oba ('A').



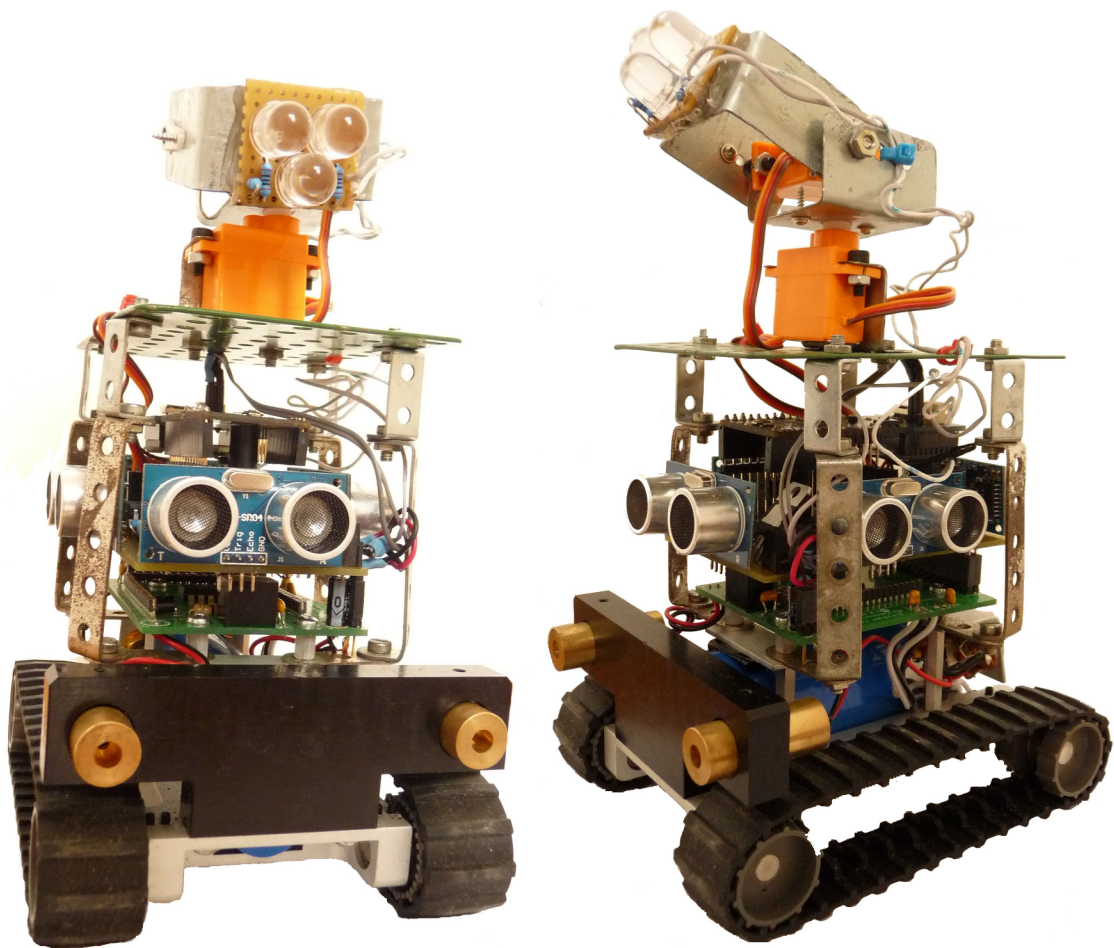
Obrázek 4.16: Pseudokód algoritmu pro řízení robota v autonomním režimu.

4.8 Zhodnocení a testování

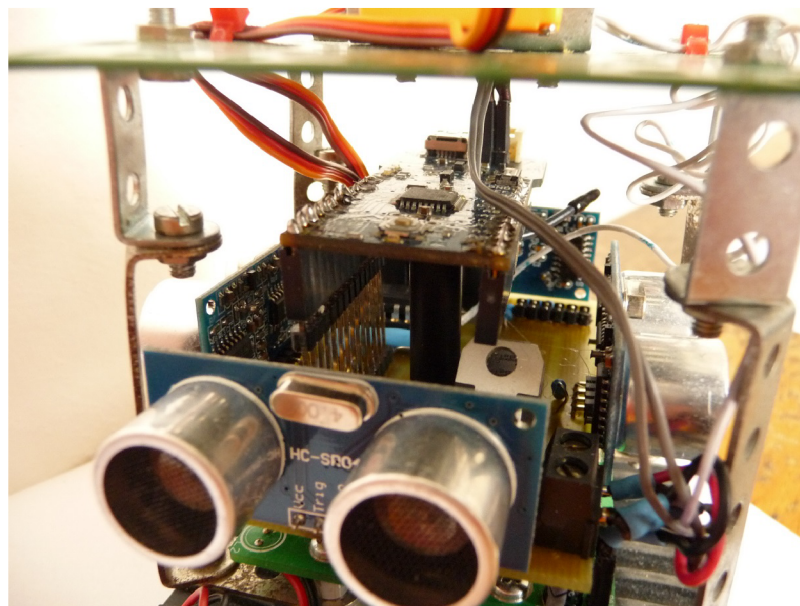
Technické nedokonalosti na podvozku vozítka způsobovaly zakřivenou dráhu robotického vozítka i v případě, kdy oba motory pracovaly na stejný výkon. Tuto chybu by bylo možné odstranit opravou zkřivených mechanických částí robota, případně použít jiný podvozek či celý robot. U měření vzdálenosti nastávaly problémy s náhodnými či nepřesnými hodnotami, způsobené nejspíše rychlým sledem měření po sobě, umocněným velkým počtem senzorů na vozítku. Mechanická konstrukce jak světelné rampy, tak usazení elektroniky v krabici u modulu ovladače je pouze ilustrativní, jelikož nebyla součástí zadání práce ani není pro prezentaci využití platformy Arduino a řízení robotického vozítka podstatná.

Při experimentech byly zjištěny výše zmíněné konstrukční nedostatky, přičemž byl zjišťován dosah signálu mezi oběma moduly v závislosti na okolí. V uzavřených prostorách byl naměřen průměrný dosah 14 m, ve volném prostranství byla naměřena maximální vzdálenost, při které již moduly přestaly komunikovat, kolem 16 m vzdušnou čarou. Při pokusu ovládat vozítko skrz zeď, případně za rohem, došlo ke ztrátě spojení při ztrátě přímé viditelnosti vozítka vůči ovladači. Rozdíly oproti teoretickým maximálním hodnotám, uvedené ve specifikaci modulů XBee Série 1, jsou způsobeny uzavřením antény uvnitř plastové krabice modulu ovladače.

Výsledný modul vozítka je zobrazen na Obrázku 4.17 a detail upevnění desky Arduino FIO k rozšiřující desce plošných spojů na Obrázku 4.18.



Obrázek 4.17: Výsledná podoba modulu robotického vozítka.



Obrázek 4.18: Detailní pohled na vnitřní usazení jednotlivých komponent modulu vozítka.

Kapitola 5

Závěr

Tato práce měla za úkol seznámit se s HW platformou Arduino a možnostmi bezdrátové komunikace. K této problematice byly nastudovány příslušné materiály a publikace, z kterých pak byly čerpány informace pro tvorbu praktické i teoretické části práce. Po výběru periférií a komponent k praktické části práce byly zhotoveny oba moduly, jak ovladač tak vozítko, s využitím prostředků fakulty.

Při implementaci bezdrátové komunikace byl implementován obousměrný přenos, kdy nejen uživatel odesílá požadavky na řízení modulu vozítka, ale i modul vozítka na tyto akce odpovídá patřičnou reakcí. Pro řízení vozítka byla využita dvojice analogových joysticků s integrovaným tlačítkem, jejichž funkce slouží k aktivaci speciálních vlastností či aktivit jak vozítka, tak ovladače. Pro ochranu desky Arduino UNO na modulu ovladače, včetně ostatní elektroniky, byl modul umístěn do plastové krabičky, spolu s napájením. Modul vozítka je osazen rozšiřující deskou, na kterou je pak připojeno Arduino FIO s XBee modulem, který je základem bezdrátové komunikace obou zařízení. Vlastním přínosem je zmíněná bezdrátová komunikace, která je navíc opatřena zabezpečujícími prvky proti výpadku signálu či jeho ztrátě pomocí párování zařízení, vlastní návrh a osazení rozšiřujících desek k oběma modulům, použití ultrazvukových senzorů pro měření vzdálenosti a samotná světelná rampa s LED diodami, která slouží pro osvětlování okolí vozítka.

Pro další rozšíření této práce by bylo nutné jisté funkce nahradit novými, jelikož jsou všechny vstupně/výstupní vývody použity a obsazeny. Z toho pramení i případná nutnost návrhu a výroby nových rozšiřujících desek pro oba moduly. Světelná rampa byla původně zamýšlena jako rampa pro kameru, která by přenášela signál na druhý modul, nicméně pro tento přenos není Arduino ani moduly XBee dostatečně výkonné a vhodné. Lze však (po zmiňovaných úpravách) práci rozšířit například o GPS senzor, a vytvořit tak plně autonomní vozítko i s možnostmi určování souřadnic, na které má vozítko zajet. Také rozhodovací algoritmus autonomní činnosti je velmi zjednodušený a slouží jako ukázka možností s použitými perifériemi. Tento algoritmus by mohl být rozšířen o získávání údajů ze všech 4 senzorů a také kvalitnějšího rozhodování o následujícím směru jízdy.

Literatura

- [1] Surveyor SRV-1 Blackfin Robot. [online], Datum poslední aktualizace: 27.04.2010, [cit. 02.12.2012].
URL http://www.surveyor.com/SRV_info.html
- [2] Arduino: Arduino.cc. [online], [cit. 10.12.2012].
URL <http://arduino.cc/>
- [3] Arduino: Arduino.cc - Arduino XBee Shield. [online], [cit. 06.12.2012].
URL <http://arduino.cc/en/Guide/ArduinoXbeeShield>
- [4] ElecFreaks: Ultrasonic Ranging Module HC – SR04. [online], [cit. 07.04.2013].
URL <http://www.micropik.com/PDF/HCSR04.pdf>
- [5] Electronics, S.: SparkFun Electronics - XBee Serie 2. [online], [cit. 07.12.2012].
URL <https://www.sparkfun.com/products/11217>
- [6] Faludi, R.: *Building Wireless Sensor Networks*. O'Reilly Media, Inc., 2010, ISBN 978-0-596-80773-3.
- [7] Igoe, T.: *Making things Talk, Second Edition*. O'Reilly Media, Inc., 2011, ISBN 978-1-449-39243-7.
- [8] Inc., D. I.: The Major Differences in the XBee Series 1 vs. the XBee Series 2. [online], [cit. 07.12.2012].
URL <http://www.digi.com/support/kbase/kbaseresultdet1?id=2213>
- [9] Kingbright: Photocoupler KB847. [online], [cit. 20.04.2013].
URL <http://www.farnell.com/datasheets/1446042.pdf>
- [10] Margolis, M.: *Arduino Cookbook*. O'Reilly Media, Inc., 2011, ISBN 978-0-596-80247-9.
- [11] OptoSupply: 10mm Round Pure White LED. [online], [cit. 07.04.2013].
URL <http://www.optosupply.com/UploadFile/PDF/osw5dka131a.pdf>
- [12] Oxeer, J.: Arduino Shield list. [online], 2011, [cit. 10.12.2012].
URL <http://www.shieldlist.org>

Příloha A

Obsah CD

V této příloze jsou na přiloženém CD disku uloženy všechny zdrojové soubory potřebné pro oba moduly a vývojové prostředí Arduino IDE verze 1.01. Také jsou zde přiloženy použité obrázky a fotografie v této písemné práci, schémata zapojení a návrhy DPS, spolu s natočenými videi pro prezentaci vozítka. Přesný popis adresářové struktury je uveden v tomto výčtu:

- **arduino/** použité vývojové prostředí Arduino IDE verze 1.01
- **fig/** obrázky a fotografie použité v písemné práci
- **pdf/** výsledný PDF dokument této práce
- **sch/FIO** schéma zapojení a návrh plošného spoje pro modul vozítka
- **sch/UNO** schéma zapojení a návrh plošného spoje pro modul ovladače
- **src/FIO** zdrojové soubory pro modul vozítka
- **src/UNO** zdrojové soubory pro modul ovladače
- **tex/** zdrojové soubory pro L^AT_EX k sestavení výsledného PDF souboru
- **vid/** prezentační videa praktické práce

Příloha B

Zdrojové kódy

Zde je zveřejněna upravená knihovna, použitá v této práci, pro ultrazvukové senzory HC-SR04, jejíž originál je zveřejněn na stránkách <http://blog.iteadstudio.com/arduino-library-for-ultrasonic-ranging-module-hc-sr04/#comment-586>.

```
1  /* Autor: Martin Čechmánek */
2  /* Datum: 2013-02-25 */
3  /* Verze: 1.0 */
4  /* Popis: knihovna pro práci s ultrazvukovými senzory pro Arduino */
5  /* Poznámka: Tato knihovna je inspirována/převzata z knihovny na
      stránkách */
6  /* http://blog.iteadstudio.com/arduino-library-for-ultrasonic-ranging-
      module-hc-sr04/#comment-586 */
7
8  #ifndef US_h
9  #define US_h
10
11 #include "Arduino.h"
12 #define TIMEOUT 5800
13
14 class Ultrasonic {
15     public:
16         Ultrasonic(int TP, int EP);
17         long zmer_vzdalenost();
18
19     private:
20         int Trig_pin;
21         int Echo_pin;
22         long vzdalenost;
23 };
24
25 Ultrasonic::Ultrasonic(int TP, int EP)
26 {
27     pinMode(TP, OUTPUT);
28     pinMode(EP, INPUT);
29     Trig_pin=TP;
30     Echo_pin=EP;
31 }
32
33 long Ultrasonic::zmer_vzdalenost()
34 {
35     digitalWrite(Trig_pin, LOW);
36     delayMicroseconds(2);
37     digitalWrite(Trig_pin, HIGH);
```

```
38   delayMicroseconds(10);
39   digitalWrite(Trig_pin, LOW);
40   vzdalenost = pulseIn(Echo_pin, HIGH, TIMEOUT);
41   return(vzdalenost/58);
42 }
43
44 #endif
```