

JIHOČESKÁ UNIVERZITA

**Přírodovědecká fakulta
Ústav aplikované informatiky**

Obor: Aplikovaná informatika



BAKALÁŘSKÁ PRÁCE

**Problematika autonomie robotů založených na
technologii Arduino**

Vedoucí práce: PhDr. Milan Novák, Ph.D.

Autor práce: Jan Hájek

České Budějovice 2016

Bakalářská práce

Jihočeská Univerzita v Českých Budějovicích

Přírodovědecká fakulta

Ústav aplikované informatiky

Autor: Jan Hájek

Název: Problematika autonomie robotů založených na technologii Arduino

Vedoucí práce: PhDr. Milan Novák, Ph.D.

Obor studia: Aplikovaná informatika – Design webových aplikací

České Budějovice 2016

ZADÁNÍ PRÁCE

Práce se bude zaměřovat na problematiku autonomních robotů, vytvořených pomocí vývojové desky Arduino. Bude obsahovat teoretickou studii možných přístupů ve formě analýzy. Na základě provedené analýzy autor práce vytvoří a otestuje získané závěry na autonomním robotnickém zařízení (např. Minisumo).

Základní doporučená literatura:

- SPONG, Mark W a Mathukumalli VIDYASAGAR. Robot dynamics and control. New York: Wiley, c1989, xvi, 336 p. ISBN 04-716-1243-X.
- ŠOLC, František a Mathukumalli VIDYASAGAR. Robotika, modelování a řízení robotů: Robotics, modelling and control of robots : teze přednášky ke jmenování profesorem v oboru "technická kybernetika". Brno: VUTIUM, 2004, 28 s. ISBN 80-214-2618-7.
- WARREN, John-David a Mathukumalli VIDYASAGAR. Arduino robotics: Robotics, modelling and control of robots : teze přednášky ke jmenování profesorem v oboru "technická kybernetika". Brno: Apress, c2011, xxiv, 601 s. Technology in action series. ISBN 978-1-4302-3183-7.

Bibliografické údaje

Hájek J., 2016: Problematika autonomie robotů založená na technologii Arduino.

[The issue of autonomy of robots based on Arduino technology. Bc. Thesis, in Czech.] – 52p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Bakalářská práce pojednává o problematice autonomních robotů vytvořených pomocí vývojové desky Arduino Uno. Obsahuje teoretickou analýzu možných přístupů k autonomním robotům. Je zde navrženo řešení pro vytvoření autonomního robotického zařízení schopného navigace pomocí systému GPS. V praktické části jsou mimo jiné popsány problémy a testování spojené s konstrukcí robota.

Klíčová slova

Autonomie, Arduino, GPS, Kompas, Wi-Fi, Robot

Annotation

This thesis discusses the problems of autonomous robots created by the development board Arduino Uno. It contains a theoretical analysis of possible approaches to autonomous robots. It proposes a solution to the creation of an autonomous robotic device capable of navigation using GPS. The practical part provides the description of problems and testing associated with the construction of the robot.

Keywords

Autonomy, Arduino, GPS, Compass, Wi-Fi, Robot

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích, dne

Podpis

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce PhDr. Milanu Novákovi, Ph.D. za ochotu a věnovaný čas při odborných konzultacích, které mi pomohli k úspěšnému vytvoření bakalářské práce. Dále Ústavu aplikované informatiky za propůjčení podvozku a dalších komponent.

Obsah

1	Úvod.....	9
2	Cíle a úkoly.....	10
2.1	Hlavní cíle	10
2.2	Úkoly.....	10
3	Metodologie (metodika)	11
3.1	Výzkumné metody	11
3.2	Výzkumné problémy.....	11
4	Úvod do problematiky	12
4.1	Autonomie robotů.....	12
4.2	Architektura robotického systému	13
4.3	Lokalizace v mobilní robotice.....	14
4.4	SLAM.....	17
5	Technologie.....	18
5.1	Arduino.....	18
5.2	Shieldy.....	19
5.3	Moduly	23
5.4	Čidla pro detekci překážek.....	29
5.5	Algoritmy	30
5.6	Komunikace mezi prostředkem a operátorem.....	35
5.7	GPS.....	36
5.8	Webová aplikace	41
6	Návrh řešení.....	42
6.1	Zapojení prvků	43
6.2	Rozdělení modulů.....	44

6.3	Algoritmizace	44
6.4	Testování	46
7	Závěr	48
8	Literatura.....	49
9	Přílohy.....	51
9.1	Seznam Obrázků.....	51
9.2	Seznam Tabulek	51
9.3	Seznam Zdrojových kódů.....	52
9.4	Struktura přiloženého CD.....	52

1 Úvod

Robotika je relativně mladý obor moderní technologie, která překračuje tradiční technické hranice. Je chápána jako „disciplína o vytváření inteligentních strojů integrujících několik vědeckých a inženýrských oblastí“ [1].

Důvodem lidského zájmu o roboty může být ověření tvůrčích schopností člověka, snaha o napodobení přírody a proniknutí do jejích zákonitostí. Ve většině případů jde ale o snahu vyrobit dokonalého pomocníka, který by dokázal totéž co my, nebo i více a to přesněji, rychleji a spolehlivěji.

Srdcem a mozkiem takového pomocníka je počítač. Počítače jsou k sehnání v různých velikostech a cenách. Pro jednoúčelové aplikace se používají takzvané mikrokontroléry. Vyznačují se velkou spolehlivostí a kompaktností. Často bývají tyto jednočipové počítače součástí vestavěných (embedded) systémů¹.

Trendem dnešní doby je použití takzvaných „low cost DIY hacking“ zařízení, což můžeme do češtiny přeložit jako „nízkonákladové udělej si sám programování z radosti či přesvědčení.“ Hlavními představiteli těchto jednodeskových počítačů je například Raspberry Pi, Arduino, Intel Edison nebo AMD Gizmo Board [2]. Tyto desky mají bohaté možnosti rozšíření o další hardware, zejména I/O moduly.

Tato práce se zaměřuje na mikrokontrolér Arduino. Protože se jedná o Open Source projekt, vytvořila se kolem Arduina velká komunita lidí, která sdílí své nápady, rady a zdrojové kódy.

¹ jednoúčelový systém, ve kterém je řídicí počítač zcela zabudován do zařízení, které ovládá. Vzhledem k tomu, že systém je určen pro konkrétní účel, mohou tvůrci systém při návrhu optimalizovat pro konkrétní aplikaci, a tak snížit cenu výrobku. [19]

2 Cíle a úkoly

Cílem této práce je vytvořit autonomního robota, který je schopný samostatného pohybu v prostoru. Ze zadání práce bylo zvoleno Arduino UNO jako řídicí jednotka. Při realizaci robota očekáváme velkou škálu problémů. Zejména pak s paměťovou náročností u řídicí jednotky, možnosti její optimalizace, přesností navigace a s tím spojené použití kompasu.

V praktické části bude návrh, který nám vyjde z teoretické části. Budou zde řešeny problémy vycházející z konstrukce robota. V neposlední řadě zde budou popsány výsledky jednotlivých měření systému

Na začátku práce stanoveny tyto cíle:

2.1 Hlavní cíle

1. Vytvoření autonomního robota schopného samostatného pohybu v prostoru
 - a. Navigace podle GPS přijímače s kompasem (bez kompasu)
 - b. Autonomní vyhýbání se překážkám
 - c. Možnost bezdrátové výměny informací mezi operátorem a prostředkem
2. Výběr vhodné navigační a lokalizační jednotky
3. Měření přesnosti navigačních a lokalizačních jednotek
4. Rozhodnout, zda je Arduino UNO vhodné pro tuto problematiku

Aby byly splněny výše uvedené cíle, musí být stanoveny pracovní úkoly. Tyto úkoly budou v průběhu práce plněny a popsány.

2.2 Úkoly

1. Studium zdrojů a literatury
2. Analýza možností pozičních systémů
3. Vytvoření webové aplikace pro vizualizaci a čtení dat
4. Realizace komunikačního mostu mezi robotem a webovou aplikací
5. Měření přesnosti výpočtu kurzu robota

6. Návrh algoritmu pro pohyb robota
7. Výběr algoritmu a čidel pro detekci překážek
8. Analýza senzorů pro detekci překážek

3 Metodologie (metodika)

3.1 Výzkumné metody

V teoretické části bude použita metoda obsahové analýzy. Tato metoda spočívá ve studiu různých zdrojů, ze kterých nám vyjdou získané závěry. Na základě této analýzy provedeme analýzu Arduina, respektive jeho technologického řešení, a definujeme jeho omezení. Z analýzy jiných řídicích systémů zjistíme výhody a nevýhody řídicí jednotky Arduino.

3.2 Výzkumné problémy

1. Je Arduino Uno R3 vhodná řídicí jednotka pro autonomního robota?
2. Jakým způsobem vyměňovat informace mezi autonomním vozidlem a operátorem?
3. Jaká čidla jsou vhodná pro detekci překážek?
4. Jaký formát (věty) použít pro vozidlo navigované pomocí GPS.

Teoretická část

4 Úvod do problematiky

4.1 Autonomie robotů

Pojmem **robot** je označován stroj, který získává informace o prostředí, ve kterém se pohybuje. Dále musí být schopen toto prostředí fyzicky, především mechanicky ovlivňovat [3]. Existuje mnoho definic robota.

*Programovatelný, multifunkční manipulátor navržený k přemísťování materiálů, nástrojů a dalších specializovaných zařízení pomocí programovatelných pohybů*².

*1. Stroj, který vypadá jako lidská bytost a provádí různé úkony jako člověk (např. chůze a řeč). 2. Automaticky řízený mechanismus*³.

*Programovatelný systém, který je schopen orientovaně vnímat a rozpoznávat prostředí, popř. manipulovat s předměty a pohybovat se*⁴.

Obecně lze stroj označit robota, pokud splňuje tyto základní vlastnosti:

- získávat a zpracovávat informace o svém okolí
- provádět určitou fyzickou činnost (pohybuje se, manipuluje s okolními předměty)

Mobilní robot je takový robot, který se dokáže sám přesouvat v prostoru. Můžeme jej rozdělit podle míry jejich autonomie na teleoperované až po plně autonomní.

Teleoperovaný robot vykonává pouze příkazy lidského operátora. Takový robot je zcela závislý na operátorovi. Některé činnosti ovšem může robot částečně vykonávat autonomně. Potom je takový robot označován za semi-autonomního. Robot, který je

² Robot Institute of Amerika, 1979

³ Merriam-Webster Collegiate Dictionary, 1999

⁴ Encyklopedie Diderot, 2002

označován za plně autonomního, je schopen vykonávat činnosti zcela sám. Popřípadě mu operátor zadá cíle a prostředek sám volí nejlepší způsob k dosažení zadaných cílů.

Autonomie (z řec. *autonomos*, řídící se vlastními zákony⁵) nebo také *funkční samostatnost*.

Autonomní robot

Aby mohl být mobilní robot označen za autonomní, musí splňovat následující vlastnosti [4] [5]:

- schopnost dlouhodobé samostatné činnosti bez nutnosti lidského zásahu
- samostatně se pohybovat pracovním prostředím
- získávat informace o pracovním prostředí
- vyhýbat se situacím, které by mohli vést ke zranění lidské obsluhy nebo zničení robota

4.2 Architektura robotického systému

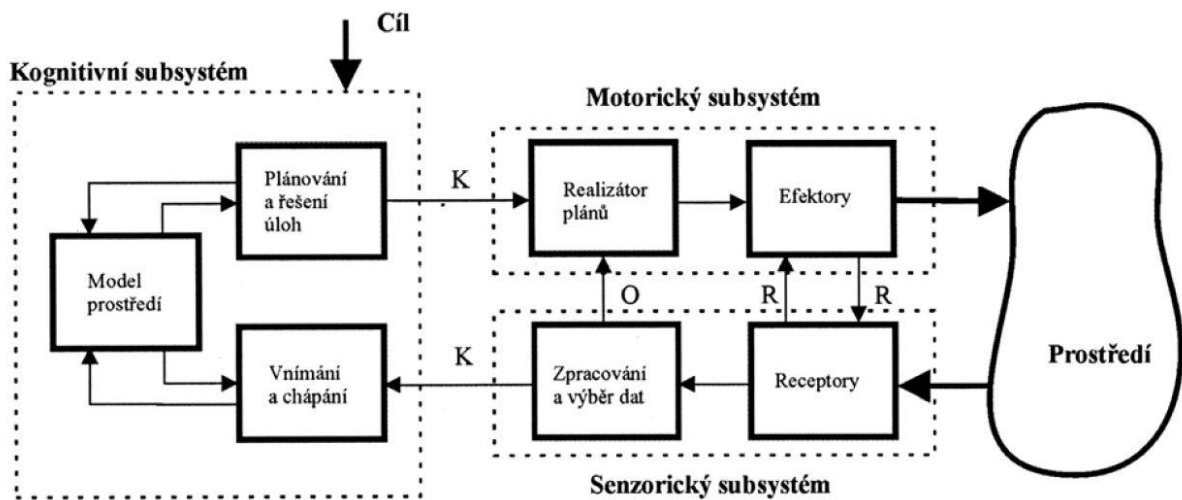
V této kapitole popíšeme základní komponenty, které tvoří mobilní roboty. Architektura robotu je jakýsi obecný návrh hardwarových a softwarových komponent, které spolu navzájem spolupracují a komunikují.

Snaha inženýrů je rozložit tuto architekturu na jednodušší části (subsystémy). Struktura a činnost subsystémů závisí na konkrétních úkolech, které má robot vykonávat. Podle blokového schématu robota Obrázek 1 můžeme vytvořit čtyři následující vrstvy:

- Motorický subsystém
- Senzorický subsystém
- Kognitivní (řídící) subsystém
- Komunikační subsystém

Pro komunikaci s operátorem lze zařadit komunikační subsystém, který předává instrukce a informace mezi řídicím subsystémem a operátorem.

⁵ Filosofický slovník. Olomouc: FIN 1998. Heslo Autonomie, str. 39.



Obrázek 1: Blokové schéma robota [3]

4.3 Lokalizace v mobilní robotice

Jeden ze základních problémů v mobilní robotice je lokalizace robota. Pro autonomního robota je nutností znát svoji polohu na základě informací ze senzorů. Je to jeden z hlavních předpokladů pro správné řešení úloh, jako je plánování cílů nebo navigace.

4.3.1 Relativní a absolutní lokalizace

Lokalizační techniky můžeme rozdělit podle charakteru měření na relativní a absolutní.

Relativní lokalizace

Pomocí relativní lokalizace můžeme odhadnout relativní změnu polohy robota. Jde především o posunutí a rotaci v rovině vzhledem k předcházející poloze. Celkovou změnu polohy určujeme skládáním jednotlivých změn poloh vůči počáteční, nebo poslední poloze. Vzhledem k nepřesnostem v jednotlivém měření dochází k chybě v odhadu polohy. Chyba se tedy zvětšuje s počtem měření. Proto je tato technika vhodná pouze pro krátkodobý odhad.

Absolutní lokalizace

Absolutní lokalizace nám umožňuje zjistit absolutní polohu prostředku v prostoru bez znalosti předchozího stavu nebo události, která vedla k dosažení dané polohy. Hlavní výhodou, oproti relativní lokalizaci, je absence akumulující se chyby. Na druhou stranu je výpočetně náročnější a může být omezená pouze na část prostoru.

4.3.2 Globální a lokální techniky

Podle nutnosti znát počáteční polohu robota a podle typu problému můžeme lokalizační techniky rozdělit na globální a lokální.

Lokální techniky

Při těchto technikách je robotovi jeho počáteční poloha v okamžiku zapnutí známá. V průběhu tyto techniky udržují odhad aktuální polohy robota a vypořádávají se s jednotlivými chybami měření. Může nastat situace, při které dojde k selhání lokalizace. Při použití pouze lokálních technik může dojít ke vzniku nezotavitelné chyby.

Globální techniky

Globální techniky jsou schopny lokalizovat zařízení i bez předchozí znalosti jeho polohy. Podle schopnosti řešit následující problémy rozdělujeme dále globální techniky na dvě kategorie.

a) **Problém probuzeného robota (wake-up robot problem)**

Tyto techniky jsou schopné lokalizovat robota bez znalosti jeho polohy v době zapnutí. Robot má informace o tom, že byl probuzen, pouze neví kde.

b) **Problém uneseného robota (kidnapped robot problem)**

Globální techniky, které řeší problém uneseného robota, dokážou lokalizovat robota po jeho neočekávaném přemístění. K přemístění došlo, když robot udržovat odhad své vlastní polohy.

Globální a lokální lokalizace závisí na senzorech, které nese robot a na způsobu dalšího zpracování.

4.3.3 Pasivní a aktivní lokalizace

Pasivní lokalizace

Pasivní lokalizace vytvoří a udržuje odhad polohy robotického zařízení na základě sensorických dat, které dostává na svůj vstup, vyhodnocuje a dále je předává vyšším vrstvám řídicí architektury. Oproti aktivní lokalizaci nemůže žádným způsobem zasahovat do ovládání prostředku.

Aktivní lokalizace

Aktivní lokalizace nám umožňuje převzít přímou kontrolu nad ovládáním robota. Většinou jen tehdy, je-li to potřeba ke zlepšení odhadu polohy.

4.3.4 Statické a dynamické prostředí

Prostředí, ve kterém se robot pohybuje, výrazně ovlivňuje lokalizaci mobilních robotů. Vzhledem k tomu, zda je prostředí v čase proměnlivé či nikoliv, můžeme ho rozdělit na statické a dynamické.

Statické prostředí

Statické prostředí je v čase konstantní a neměnné. Jediný pohybující se objekt je zde robot, kterého chceme lokalizovat. V tomto prostředí se neřeší možnost zákrytu robota dalším pohybujícím se objektem. Tyto konstantní vlastnosti statického prostředí slouží především pro výzkumné účely lokalizačních technik. V reálném světě, ale na takové prostředí nenarazíme.

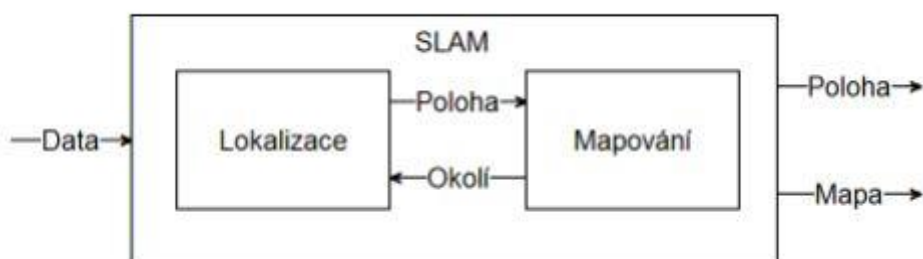
Dynamické prostředí

Vlastnosti dynamického prostředí se mění v čase, tím ovlivňuje sensorické vnímání robota. Lokalizace v proměnném prostředí představuje velký problém robotiky jako

takové. Vlastnosti, které se mění v čase, může být spousta. Například pohybující se roboti, cizí objekty, lidé, otevřené a zavřené dveře nebo úroveň osvětlení.

4.4 SLAM

Zkratka SLAM je anglický výraz „Simultaneous localization and mapping“, což do češtiny můžeme přeložit jako „Souběžná lokalizace a mapování“. Jedná se o koncept zpracování dat. Tato úloha spočívá v tvorbě mapy prostředí bez nutnosti předem daných informací a současné lokalizace. Robot z dat, které naměří jeho senzory, vytváří (aktualizuje stávající) mapu a zároveň také určuje, kde se na mapě nachází [6].



Obrázek 2: Schematické znázornění SLAM algoritmu [6]

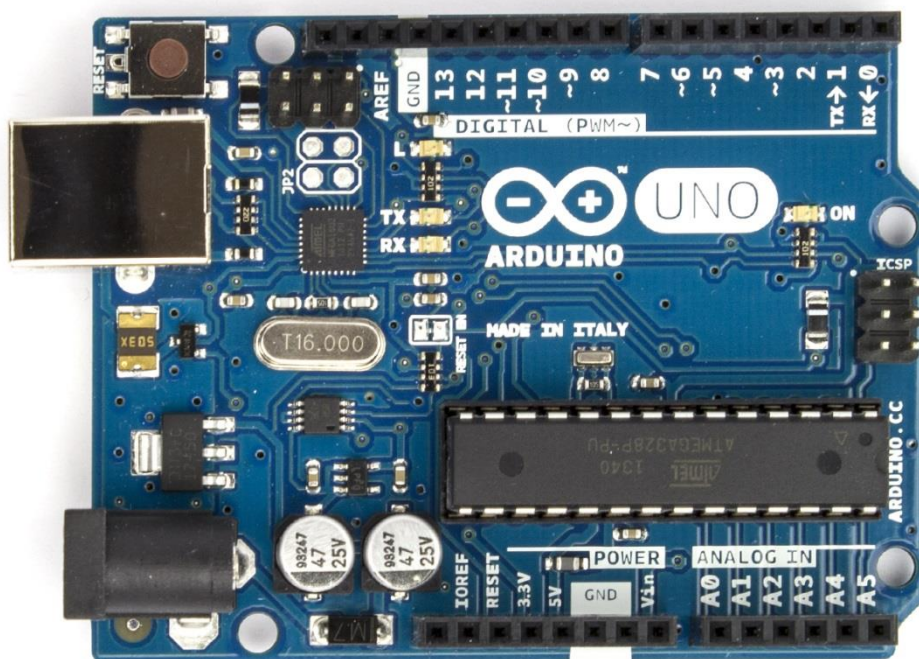
Z obrázku Obrázek 2 je vidět, že SLAM je rozdělen na dvě části, a to část lokalizace a část mapování. Pro správnou funkci je nutné zajistit fungování obou těchto částí. V okamžiku, kdy jedna část selže, předá části druhé chybná data. Ta je špatně vyhodnotí a pošle opět chybný výsledek první části.

Praktická část

5 Technologie

5.1 Arduino

Arduino je otevřená elektronická platforma, která je založena na uživatelsky jednoduchém hardware a software. Umožňuje snadný návrh a vývoj elektronických programovatelných zařízení. Základem desky je 8 bitový AVR či 32 bitový mikrokontrolér od firmy Atmel. Arduino poskytuje vstupně-výstupní digitální piny a analogové piny. Dále zde najdeme několik diod, resetovací tlačítko, napájecí konektor a oscilátor. Deska je k počítači připojena pomocí rozhraní USB, ale je zde softwarově simulovaná sériová komunikace přes linku RS-232.



Obrázek 3: Arduino UNO R3

V této práci byla použita deska Arduino UNO R3. Zde jsou technické parametry:

Mikroprocesor

- Atmel ATMega328
- 16MHz takt
- 32kB flash paměť
- 2kB SRAM paměť
- 1kB EEPROM paměť

Napájení

- 5V USB
- 7-15V adaptér (baterie)

I/O piny

- 14 digitálních
- 0,1 – sériová linka
- 2,3 – externí přerušení
- 3,5,6,9,10,11 – PWR
- 10,11,12,13 – SPI (s knihovnou SPI library)
- 4,5 – I2C (s knihovnou Wire library)
- 6 - vstupních analogových

5.2 Shieldy

Každá deska má většinu vstupně-výstupních pinů přístupných přes patice, do kterých se jednoduše připojují další obvody, kterým se říká Shieldy. Platforma Arduino umožňuje využívat tyto nadstavbové desky, které poskytují efektivní způsob, jak vybudovat odpovídající konstrukci robota.

Jak vyplívá z obecného návrhu v kapitole 6. konstrukce vyžaduje ovládání DC motorů a to za použití regulátorů. Konstrukčně lze využít H-můstky, které jsou reprezentovány integrovaným obvodem.

V první verzi prostředku byl použit H-můstek s integrovaným obvodem L293D. Tato varianta navíc umožňuje i reverzaci směru. Na čipu se nachází ochranné diody, které chrání výkonové tranzistory uvnitř, před poškozením špičkami indukovaného napětí.

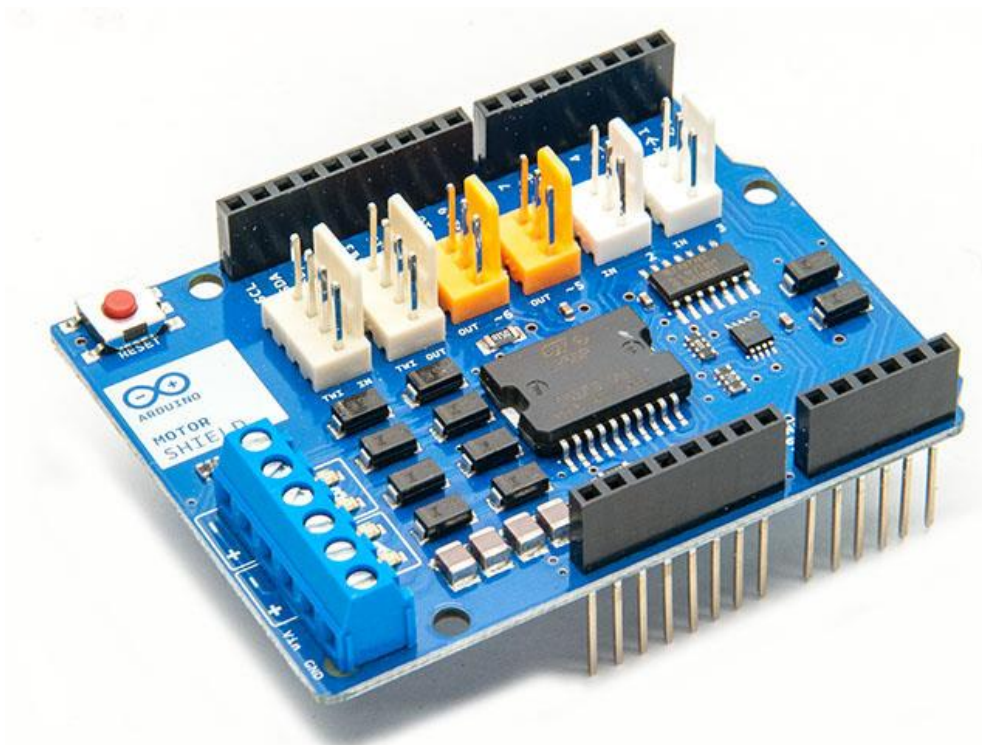


Obrázek 4: Integrovaný obvod L293D

5.2.1 Motor Shield

Další možností je využít nadstavby pro Arduino v podobě Motor Shieldu. Tento Shield má 2 kanály, což nám umožňuje řízení dvou stejnosměrných motorů. Ovládat se nechají i relé, solenoidy a krokové motory. Můžeme také napájet tento shield externím napájením až 12V a 2A na jeden motor. Mezi jeho hlavní přednosti patří především jednoduchá instalace k Arduino a to pouze zasunutím do příslušných pinů. Pomocí shieldu můžeme jednoduše zvolit kanál motoru, určit polaritu motoru a tím pádem směr. Dále je možné regulovat otáčky motoru (PWM) a zastavení nebo spuštění motoru. Mezi zajímavou vlastnost patří také sledování vytižení kanálů.

Pomocí tohoto shieldu také jednoduše vyřešíme napájení celého prostředku. Shield má provozní napětí 5-12V.



Obrázek 5: Arduino Motor Shield R3

V následující tabulce je uvedeno zapojení pinů k Arduino desce.

Funkce	Kanál A	Kanál B
Směr	D12	D13
Otáčky (PWM)	D3	D11
Brda	D9	D8
Aktuální průzkum	A0	A1

Tabulka 1: Motor Shield, zapojení pinů k Arduino

5.2.2 Wi-Fi shield

Wi-Fi Shield je další z řady oficiálních shieldů pro Arduino. Umožňuje nám připojit se k síti bez nutnosti připojování Ethernet kabelu. Tento shield využijeme pro příjem GPS souřadnic z webové aplikace. Také pomocí něho budeme posílat zpět data o aktuální poloze, směru, popř. detekované překážce.

Wi-Fi se dokáže připojit k sítím se standardem 802.11b a 802.11g a zabezpečením WEB a WPA2 Personal. Aby bylo možné se připojit k síti, je nutné, aby síť vysílala svoje SSID. Testování shieldu ukázalo, že není vhodné používat pin 7. Výrobce uvádí, že je využíván knihovnou pro ovládání shieldu.



Obrázek 6: Arduino Wi-Fi Shield

5.2.2.1 Zjištění verze firmware

Aby shield pracoval korektně, je nutné zjistit verzi firmware. Verzi zjistíme pomocí jednoduchého kódu, viz Přílohy. Verze by měla být 1.1.0 (nebo vyšší). Zdali tomu tak není, je nutné provést aktualizaci firmware.

5.2.2.2 Připojení Wi-Fi shieldu

Při porovnání tabulky 1. a zapojení Wi-Fi shieldu vyplývá, že není možné zapojit tyto dva shieldy na sebe. Překrývají se totiž piny na ovládání motoru a komunikace prostřednictvím Wi-Fi.

Funkce	Arduino pin
SCK	D13
MISO	D12
MOSI	D11
SS for Wi-Fi	D10
Handshake mezi shieldem a Arduinem	D7
SS pro SD card	D4

Tabulka 2: Zapojení Wi-Fi shieldu

5.3 Moduly

Do této kategorie můžeme zařadit vše ostatní, co lze připojit k Arduinu. Od nejrůznějších čidel, jako je například akcelerometr, kompas, detektor alkoholu nebo zvuku. Dále to může být LCD display, GSM modul, Wi-Fi modul nebo GPS modul, který si blíže popíšeme.

5.3.1 GPS modul

Při stavbě robota byl použit model Ultimate GPS Breakout v3 od firmy Adafruit. Tento model postrádá externí anténu, se kterou je možné přesnější a rychlejší zachycení signálu. GPS modul komunikuje přes sériové rozhraní RX a TX. Nyní nám stačí modul připojit k Arduinu a nahrát jednoduchý kód s knihovnamy.

aktuální pozice robota, a cílové souřadnice zadané operátorem. Výstupem funkce *courseTo()*; je tedy úhel ve stupních.

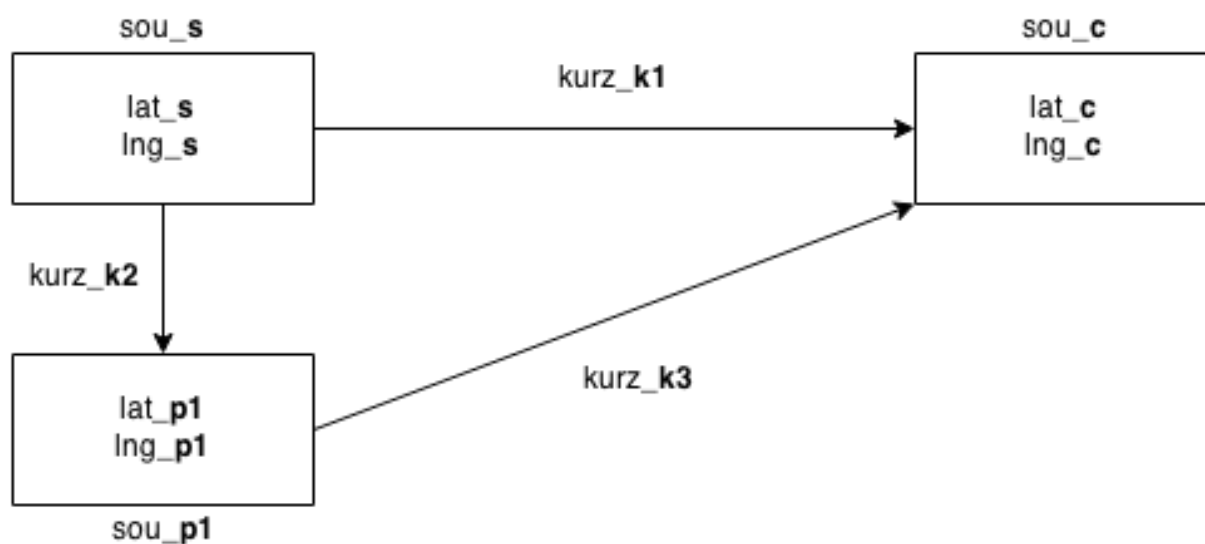
Kurz do cíle již známe a nyní je potřeba zjistit, jaký kurz má naše robotické zařízení. Toho lze docílit dvěma způsoby. Jako první možnost je zde výpočet aktuálního kurzu z pohybu vozidla.

```
double TinyGPSPlus::courseTo(double lat1, double long1, double
lat2, double long2)
{
    double dlon = radians(long2-long1);
    lat1 = radians(lat1);
    lat2 = radians(lat2);
    double a1 = sin(dlon) * cos(lat2);
    double a2 = sin(lat1) * cos(lat2) * cos(dlon);
    a2 = cos(lat1) * sin(lat2) - a2;
    a2 = atan2(a1, a2);
    if (a2 < 0.0)
    {
        a2 += TWO_PI;
    }
    return degrees(a2);
}
```

Zdrojový kód 1: Výpočet kurzu v knihovně TinyGPS++ [7]

Kurz pohybujícího se prostředku zjistíme pomocí následujících kroků. Grafické znázornění vidíme na Obrázek 8. Ze souřadnic *sou_s* a *sou_c* vypočítáme kurz *kurz_k1*. Vozidlo se nyní rozjede směrem dopředu a dorazí na souřadnice *sou_p1*, které jsou náhodné, protože vozidlo ještě nemá informace o tom, jaký je jeho kurz. Následně vypočítá *kurz_k2* ze souřadnic *sou_s* a *sou_p1*. Z tohoto vyplývá, že *kurz_k2* je nyní kurz

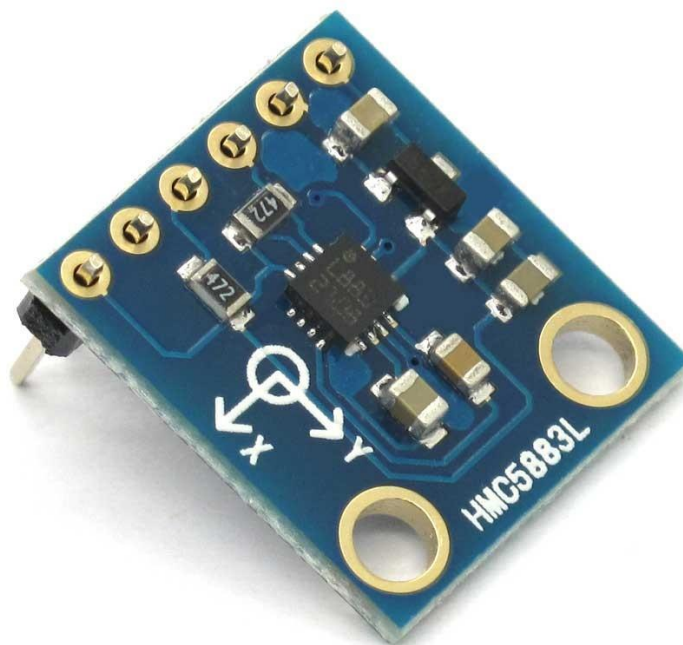
robotu. Nyní už jen stačí vypočítat *kurz_k3* ze souřadnic *sou_p1* a *sou_c* a srovnat kurz robotu *kurz_k2* a kurz do cíle *kurz_k3*.



Obrázek 8: Blokové schéma výpočtu kurzu z pohybu prostředku

Druhou možností je použití trojosého digitálního magnetometru. Jeho hlavní výhodou je znalost svého natočení vzhledem k terénu hned po zapnutí prostředku. Tím tedy odpadá nutnost pohybu pro zjištění kurzu vozidla.

5.3.2 Magnetometr HMC5883L (kompas)



Obrázek 9: Trojosý magnetometr HMC5883L

Magnetometr měří ve třech osách složku lokálního magnetického pole, které je tvořeno součtem geomagnetického pole. Pomocí něho můžeme vytvořit elektronický kompas. Magnetometry své uplatnění naleznou například v leteckém provozu, mobilních telefonech nebo GPS navigacích.

Princip:

Měření magnetického pole je založeno na tzv. magnetorezistivním jevu, při kterém se mění odpor magnetického materiálu v závislosti na změně působícího magnetického pole. Použitý 3-osý magnetometr obsahuje tři snímače magnetického pole ve formě Wheatstonových můstků složených z rezistorů a magnetorezistivního materiálu. Každý

jeden můstek je orientován ve směru měřící osy (x, y, z). Mezi výstupy můstků je po připojení napájení úměrné napětí směru a velikosti magnetického pole. [8]

Magnetometr je velmi citlivý na okolní kovové předměty a může docházet ke zkreslení. Je nutné ho proto odstínit od všeho, co by ho mohlo rušit. Čidlo komunikuje přes I²C (multi-masterová sériová sběrnice), takže je zapojení velmi jednoduché.

5.3.3 Elektromagnetická kompatibilita

Elektromagnetická kompatibilita (EMC) je schopnost magnetického nebo elektrického přístroje vykazovat správnou činnost v prostředí, kde na toto zařízení působí elektromagnetické signály. V druhé řadě se jedná o schopnost neovlivňovat svou elektromagnetickou činností své okolí. Tyto vlastnosti můžeme rozdělit do dvou kategorií: [9]

1. **Elektromagnetická susceptibilita** (EMS) – Vyjadřuje schopnost zařízení pracovat bez poruch nebo s přesně definovanou mírou rušení.
2. **Elektromagnetická interference** (EMI) – Jedná se o proces, při kterém se signál, který je generovaný zdrojem ve stejném prostředí přenáší do rušivých systémů.

Tyto základní vlastnosti EMC zahrnují společné kroky a postupy. Jedním z těchto kroků je velmi obsáhlá a důležitá oblast a tou je měření elektromagnetické interference.

U mobilního prostředku bylo nutné změřit, jak vysoko musí být umístěn kompas nad zařízením (podvozek, motory, akumulátor, wi-fi modul), protože elektromagnetický kompas modul vykazoval nekorektní data při přiblížení k prostředku. Měření na mobilním prostředku probíhalo následujícím způsobem. Čidlo kompasu bylo připojeno k desce postupně oddalováno směrem vzhůru. Na přiloženém metru se kontrolovala vzdálenost.

Hodnoty z čidla se ustálily ve vzdálenosti **15 cm** nad zařízením. Při konstrukci je proto nutné vyzdvihnout kompas modul nad zařízení minimálně do této vzdálenosti aby nedocházelo k rušení.

5.4 Čidla pro detekci překážek

Jako čidla (nebo také snímače či senzory) jsou obecně označována technická zařízení, určená pro detekci a snímání fyzikálních nebo technických veličin. Ty poté převádí na signál, který lze dále zpracovat a vyhodnocovat v řídicích a jiných systémech. Pro detekci překážek jsou zvláštním způsobem použity čidla pro měření vzdálenosti.

Čidla můžeme rozdělit podle dvou základních skupin. [10]

Množství dat

Podle množství dat, které lze daným čidlem získat na výstupu. Zde může být pouze jedna hodnota, jako je například vzdálenost nejbližšího objektu ve směru čidla. V tom nejjednodušším případě je získána pouze binární hodnota o přítomnosti nebo nepřítomnosti objektu. Patří sem například kapacitní, indukční nebo taktilní senzory. Tyto čidla se často používají u průmyslových zařízení.

U komplexnějších čidel je na výstupu soubor většího množství informací. Může to být řádková či dvourozměrná matice vzdáleností. S větším počtem informací roste i paměťová a časová náročnost zpracování.

Nicméně při větším množství dat je možné lépe analyzovat prostředí a zpřesnit plánování trasy.

Fyzikální princip

Při použití těchto principů je možnost několika způsobu měření. Jeden z nich je vysláním impulzu a čekáním na odražený způsob. Je možné také měřit fáze odraženého signálu nebo pomocí triangulační metody, která pracuje s laserovým paprskem a maticovým senzorem.

Dále můžeme snímače rozdělit podle snímaného úhlu, který může být bodový, nebo může mít velký rozptyl.

5.5 Algoritmy

Výběr vhodného algoritmu má zásadní význam pro následnou spolehlivost a dosažené výsledky při detekci překážek. Algoritmy můžeme rozdělit do několika kategorií právě podle míry spolehlivosti, rychlosti a způsobu zpracování dat.

Základní rozdělení může být na jednoduché algoritmy, které nevytvářejí žádnou nebo velmi jednoduchou mapu okolí, a složitější algoritmy, které vyhledávají optimální trasu. U složitějších (grafových) algoritmů můžeme dosáhnout přesnějších výsledků za cenu složitosti a výpočetního výkonu.

5.5.1 Jednoduché metody

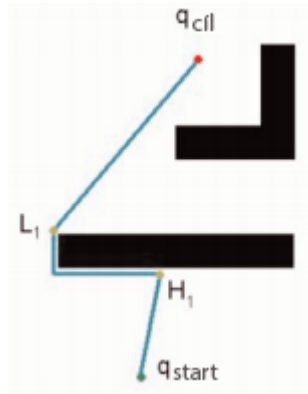
Tyto metody jsou v základu podobné. Od startu postupují k cíli to té doby, než narazí na překážku. V této chvíli není možné pokračovat do zvoleného směru a musí být stanovena náhradní cesta. Můžeme sem zařadit například Bug algoritmy.

Bug algoritmy

Jsou to jedny z nejjednodušších algoritmů pro obcházení překážky. Aby bylo možné použít tyto algoritmy, je nutné znát polohu robota a přesně změřené vzdálenosti k překážkám. [10]

Bug 0

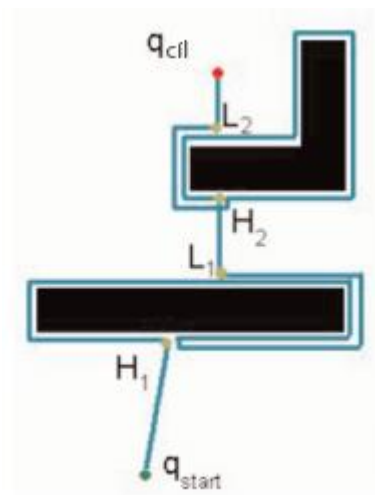
Při aplikaci tohoto algoritmu se robot vydá ze své startovací pozice přímo k cíli. V případě, že narazí na překážku, začne jí objíždět ve zvoleném směru (pravidlo, pravé nebo levé ruky). Pokud překážka neprotíná spojnicí mezi svou polohou a cílem, robot může pokračovat v cestě.



Obrázek 10: Algoritmus Bug 0

Bug 1

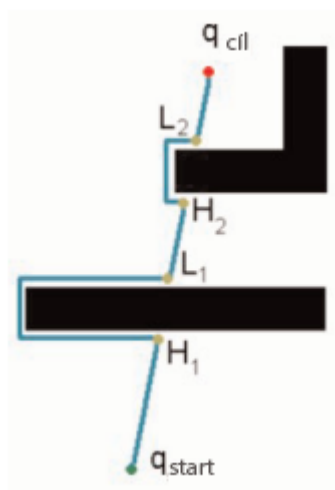
Stejně jako při algoritmu Bug 0 se robot vydá přímo do cíle, než narazí na překážku. Při střetu s překážkou si tento bod zapamatuje (h_1) a začne překážku obcházet v určeném směru. Zde však robot obejde celou překážku, než narazí na bod h_1 . Po dobu obcházení řídicí jednotka robotu počítá vzdálenost k cíli. Po objetí vyhodnotí nejmenší vzdálenost k cíli a směr, kterým se dostane z bodu h_1 do bodu s nejkratší vzdáleností k cíli a přesune se k němu. Z tohoto bodu směřuje přímo k cíli.



Obrázek 11: Algoritmus Bug 1

Bug 2

Bug 2 se od předcházejících algoritmů liší v tom, že pamatuje spojnicí startu a cíle po celou dobu obcházení překážek. Začne se tedy přesouvat ze startovní pozice do cíle, než narazí na překážku. Překážku začne obcházet, dokud nenarazí na uloženou spojnicí. Pokud je tento bod blíže k cíli, pokračuje po spojnicí dále.



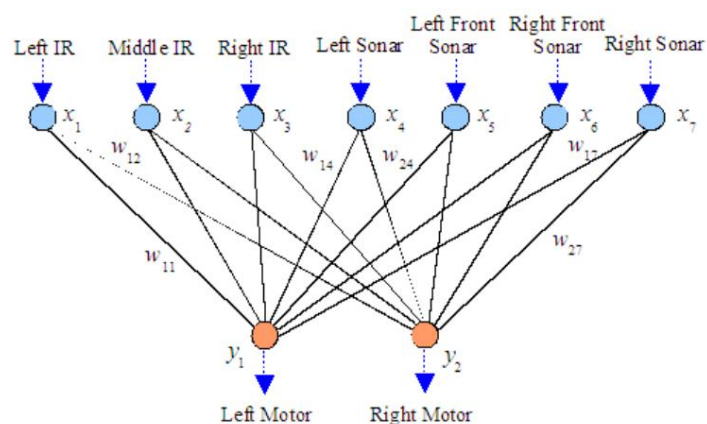
Obrázek 12: Algoritmus Bug 2

Existuje další řada algoritmů, které jsou kombinací předešlých.

5.5.2 Umělé neuronové sítě

Umělé neuronové sítě byly navrženy podle funkčních buněk jednoduchých neuronů. Tyto sítě mají široké uplatnění právě v algoritmizaci. Po naučení dokáží velmi efektivně a spolehlivě pracovat v naučeném rozsahu vstupních dat. Jedinou nevýhodou je potřeba učení na trénovacích datech. Mimo tyto data se může systém chovat nepředvídatelně. Díky učení z trénovacích dat není nutná přesná znalost systému robota.

Pokud budeme od neuronových sítí vyžadovat pouze jednoduchý úkol jakým je vyhnutí se kolizi, bude neuronová síť jednoduchá, jednovrstvá a bude obsahovat pouze několik neuronů. Data bude sbírat malého počtu senzorů pro měření vzdálenosti. Na výstupu sítě může být údaj pro natočení podvozku.



Obrázek 13: Neuronová síť pro vyhýbání se překážkám

Při požadavku nalezení nejkratší možné trasy by se problém stal mnohem složitější. Neuronová síť by musela obsahovat paměť, aby nedošlo k zacyklení. Struktura takové sítě by nebyla jednoduchá, učení náročné a zdlouhavé. [11]

5.5.3 Ultrazvukový senzor HC-SR04

Technické parametry:

Pracovní napětí 5V.

Pracovní proud 15mA.

Rozsah snímání 2 - 400cm.

Měřicí úhel 15°.

Princip

Ultrazvukový senzor vyvolává vysokofrekvenčně zvukové vlny a zachytává echo, které je získané odrazem od překážky. Elektronika senzoru vypočítá časový interval mezi vyslaným a přijmutím signálem. Takto lze získat vzdálenost od objektu.



Obrázek 14: Ultrazvukový senzor HC-SR04 [12]

5.5.4 IR čidlo vzdálenosti Sharp 2Y0A02

Technické parametry:

Pracovní napětí 5V.

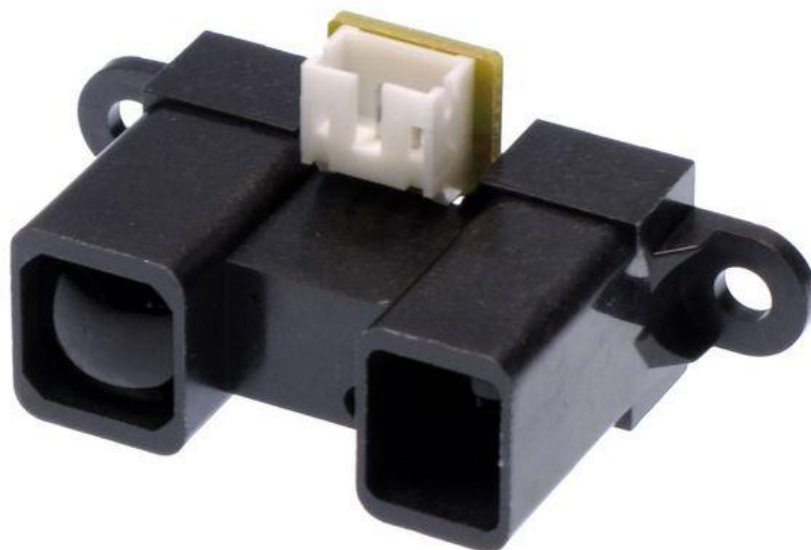
Pracovní proud 30mA

Rozsah snímání 10 – 80cm.

Princip

Infračervený senzor pro měření vzdálenosti pracuje na podobném principu jako zařízení pro přenos dat. Vyslaný paprsek se odrazí od překážky a je zachycen na přijímací části.

Ta jej zaregistruje a vyhodnotí. [13]



Obrázek 15: Čidlo vzdálenosti Sharp 2Y0A02

5.6 Komunikace mezi prostředkem a operátorem

Pro nastavení souřadnic prostředku máme několik možností. Tou první je nastavit souřadnice takzvaně „natvrdo“ přímo do kódu. Je to ta nejjednodušší varianta, ale má své nevýhody. Souřadnice nelze během provozu měnit a je nutné připojení kabelu (typ AB). Také není možno aktuální stav prostředku sledovat.

Z těchto důvodů byla navržena webová aplikace, které zprostředkovává komunikaci mezi Arduino deskou a operátorem. K desce je připojený již zmíněný wi-fi shield od firmy Adafruit Industries.

5.6.1 aREST knihovna

Jedná se o jednoduchou knihovnu, která implementuje REST API pro Arduino a wi-fi čip ESP8266. Je univerzální a v současné době podporuje RESTové volání prostřednictvím http protokolu (za použití CC3000 wi-fi chipu, Arduino wi-fi knihovny, nebo Ethernet shieldu). Podporuje také komunikaci pomocí sériového portu (za použití USB sériového spojení, Bluetooth a XBee). Pro implementaci knihovny je zapotřebí použít nejnovější verzi Arduino IDE (v současné době verze 1.6.8). [14]

REST (Representational State Transfer) - jedná se o architekturu rozhraní, která je navržena pro distribuované prostředí. REST na rozdíl od SOAP je orientovaný datově (SOAP je protokol pro výměnu zpráv založených na XML přes síť a je orientovaný procedurálně). Webové služby definují vzdálené procedury a protokol pro jejich volání a REST určuje, jak se bude přistupovat k datům. [15]

Knihovna byla otestována, ale pro toto řešení byla nevhodná z důvodu omezení posílání textových zpráv (datového typu string).

5.7 GPS

Znalost pozice je velice důležitá nejenom v robotice, ale i například ve vojenství. Z tohoto důvodu byl americkou armádou vyvinut systém pro lokalizaci označovaný jako GPS (Global Position System).

Globální družicový polohový systém je služba, umožňující za pomoci družic autonomní prostorové určování polohy s celosvětovým pokrytím. Používají se malé elektronické rádiové přijímače, které na základě odeslaných signálů z družic umožňují vypočítat jejich polohu. [16]

Tento systém má dvě třídy přesnosti:

1. **PPS** (přesná polohová služba), která autorizovaným uživatelům poskytne plnou přesnost systému. Mezi autorizované uživatele patří armáda USA, armády NATO a některých dalších států.
2. **SPS** (standartní polohová služba), která je dostupná všem uživatelům po celém světě.

Vlivů na výslednou přesnost určení polohy jsou desítky. Zde je výběr těch hlavních:

1. Konfigurace družic nad místem pozorování – Pro bezproblémové určení polohy potřebujeme vidět alespoň 4 družice. Čím je družic více, tím se zvedá přesnost a rychlost určení polohy.
2. Vliv atmosféry – Jedno z největších omezení přesnosti systému plyne z průchodu signálu atmosférou (zejména ionosférou), kde je chyba až 30m a troposférou, chyba až 3m. Tyto chyby se dají matematicky a vhodným měřením odstranit.
3. Stav družic systému – V navigační zprávě, která putuje spolu se signálem od každé družice je vysílána zpráva o tom, zda je možné družici zahrnout do výpočtu. Může se jednat o údržbu družic, korekci drah nebo testování. V těchto případech družice neposkytují kvalitní data a přijímač družici z výpočtu vyloučí.
4. Vícecestné šíření signálu – Signál, který putuje do přijímače, se cestou odrazí od jiných překážek a transitní čas je poté zkreslen. Nejvíce se tato chyba projevuje u družic letících nízko nad horizontem a velmi špatně se tato chyba detekuje. [17]

Formát NMEA-0183

NMEA – Národní asociace pro námořní elektroniku (National Marine Electronics Association) vytvořila standard, který definuje rozhraní mezi počítačem a GPS přijímačem.

Označení NMEA-0183 znamená, že data jsou posílána po řádcích.

Zprávy získané z modulu se nazývají věty. Každá věta začíná znakem dolar (\$) a pokračuje až do konce řádku. Následující dva znaky identifikují vysílač (GP = GPS) a další 3 znaky typ zprávy. Každý řádek je ukončen hvězdičkou a hexadecimálně zapsaným kontrolním součtem. Délka řádku je omezena na maximálně 80 znaků.

RMC věty

Tato zkratka je označením pro *Recommended minimum specific GPS/Transit data*. Pod touto zkratkou si můžeme představit jakési minimum, které by měla poskytovat většina GPS přijímačů.

Příklad:

\$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A

#	příklad	komentář
1	RMC	Recommended Minimum sentence C
2	123519	Určuje čas, 12:35:19 UTC
3	A	Status (A = Active, V = Void) platná / neplatná pozice
4	4807.038,N	Latitude: 48 stupňů 07.038 minut, severní šířky
5	01131.000,E	Longitude: 11 stupňů 31.000 minut, východní délky
6	022.4	Rychlost v nadmořských uzlech
7	084.4	Azimut ve stupních
8	230394	Datum, 31. března 1994
9	003.1,W	Magnetický rozptyl a jeho směr (západ)
10	*6A	Kontrolní součet

Tabulka 3: Popis RMC věty [18]

GGA věty

Druhou nejpoužívanější větou je GGA (*Global positioning System Fix Data*). Oproti větě RMC obsahuje navíc počet satelitů.

Příklad:

\$GPGGA,170139.615,4912.2526,N,01635.0378,E,1,07,1.0,357.5,M,43.5,M,0.0,0000*7D

#	příklad	komentář
1	170139.615	Čas (UTC), pro který platí údaje o vypočtené pozici
2	4912.2526	Zeměpisná šířka
3	N	Indikátor severní/jižní šířka (N=sever, S=jih)
4	01635.0378	Zeměpisná délka

5	E	Indikátor východní/západní délky (E=východ, W=západ)
6	1	0 - nebylo možno určit pozici, 1 - GPS, 2 – DGPS (diferenční GPS)
7	07	Počet viditelných satelitů 00 — 12
8	1.0	Vliv rozestavení družic na určení polohy HDOP (Horizontal Dilution of precision)
9	357.5	Výška antény nad geoidem
10	M	Jednotka pro předchozí údaj (č.9) (M=metr)
11	43.5	Geoidal separation, rozdíl mezi WGS-84 zemským elipsoidem a střední úrovní moře (geoid). Znaménko mínus znamená, že střední úroveň země je pod elipsoidem.
12	M	Jednotka vzdálenosti pro předchozí položku (č.11) (M=metr)
13	0.0	Stáří poslední aktualizace DGPS. Údaj je uváděn v sekundách. Jestliže údaj chybí, nepoužívá se DGPS.
14	0000	Identifikační číslo referenční stanice pro DGPS (0000 — 1023)
15	7D	Kontrolní součet

Tabulka 4: Popis GGA věty [18]

GSA věty

Věta GSA (*GPS DOP and Active Satellites*) poskytuje informace o seznamu satelitů.

Příklad:

*\$GPGSA,A,3,04,05,,09,12,,,24,,,,2.5,1.3,2.1*39*

#	příklad	komentář
1	GSA	Satellite status
2	A	Automatický výběr 2D nebo 3D mód
3	3	(M = Manual, 1 = chybí pozice, 2 = pouze 2D pozice)

4	04,05...	Kódy viditelných satelitů (max. 12)
5	2.5	Chyba pozice
6	1.3	Horizontální chyba
7	2.1	Vertikální chyba
8	*39	Kontrolní součet

Tabulka 5: Popis GSA věty [18]

GSV věty

Síla signálu je ukryta ve větě GSV (*GPS satellites in View*).

Příklad:

*\$GPGSV,2,1,08,01,40,083,46,02,17,308,41,12,07,344,39,14,22,228,45*75*

#	příklad	komentář
1	GSV	Satellites in view
2	2	Celkový počet zpráv
3	1	1. zpráva ze 2.
4	08	Počet satelitů ve výhledu
5	01	PRN číslo satelitu
6	40	Elevace ve stupních
7	083	Azimut ve stupních
8	46	Síla signálu
9	Následují maximálně 4 satelity, stejný význam jako u předešlé čtveřice	
10	*75	Kontrolní součet

Tabulka 6: Popis GSV věty [18]

Výběr věty

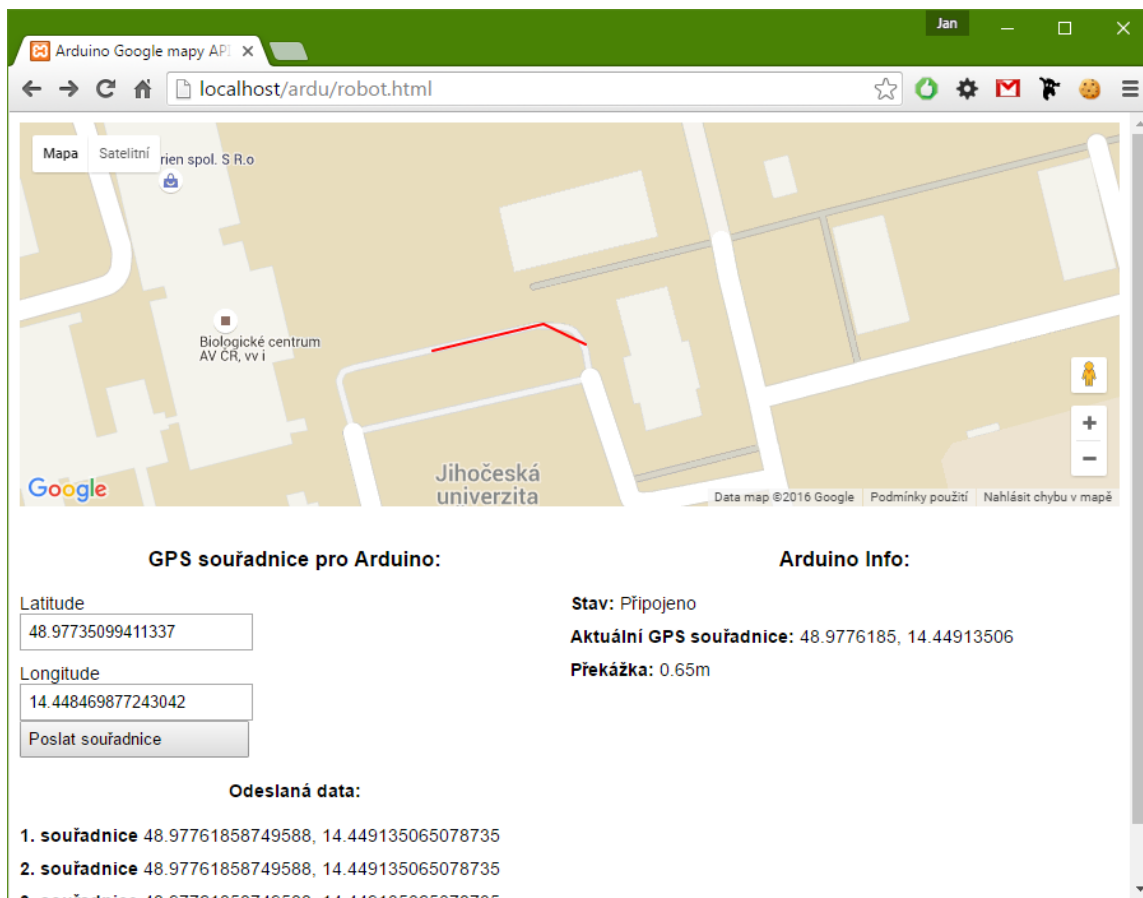
Každá věta poskytuje určité informace. Pro robotické zařízení je potřeba vybrat správnou větu, která bude splňovat všechny potřebné parametry a informace. Byly stanoveny následující požadavky.

Ta nejzákladnější podmínka je aktuální poloha, tzn. latitude a longitude. Dále budeme potřebovat počet viditelných satelitů pro přesnost měření. Z tohoto nepřeborného množství vět byla vybrána věta GGA. Tato věta obsahuje požadované informace a zároveň je podporována knihovnou TinyGPS++.h což je knihovna pro práci s těmito větami.

5.8 Webová aplikace

Webová aplikace využívá Google Maps API pro jednoduché zjištění GPS souřadnic pouhým kliknutím na mapu. Tyto souřadnice jsou poté odeslány TCP/IP protokolem do zařízení a dále zpracovány. Jako odpověď jsou přijata data o aktuální poloze, stavu zařízení a vzdálenosti od překážky. Ty se následně asynchronně zobrazí do mapy a panelu informací. Odeslání prázdného požadavku vrátí stejné informace jako při odpovědi na odeslané souřadnice.

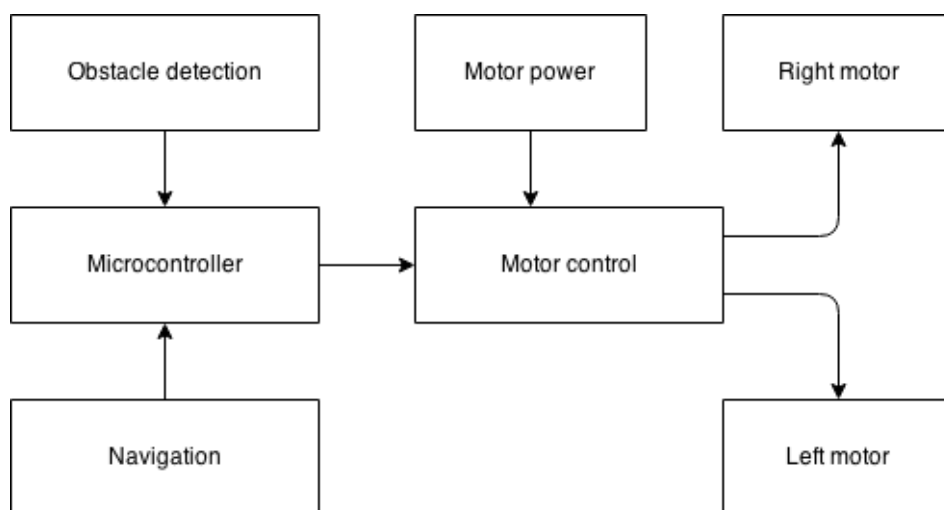
Komunikace mezi aplikací a zařízením není zabezpečena. Řešením je zabezpečení samotné wi-fi pomocí hesla. V tomto případě není řešení bezpečnosti přenosu prioritou. V případě, že by byl prostředek použit na jiný úkol, je nutné otázku bezpečnosti přenosu řešit.



Obrázek 16: Webová aplikace

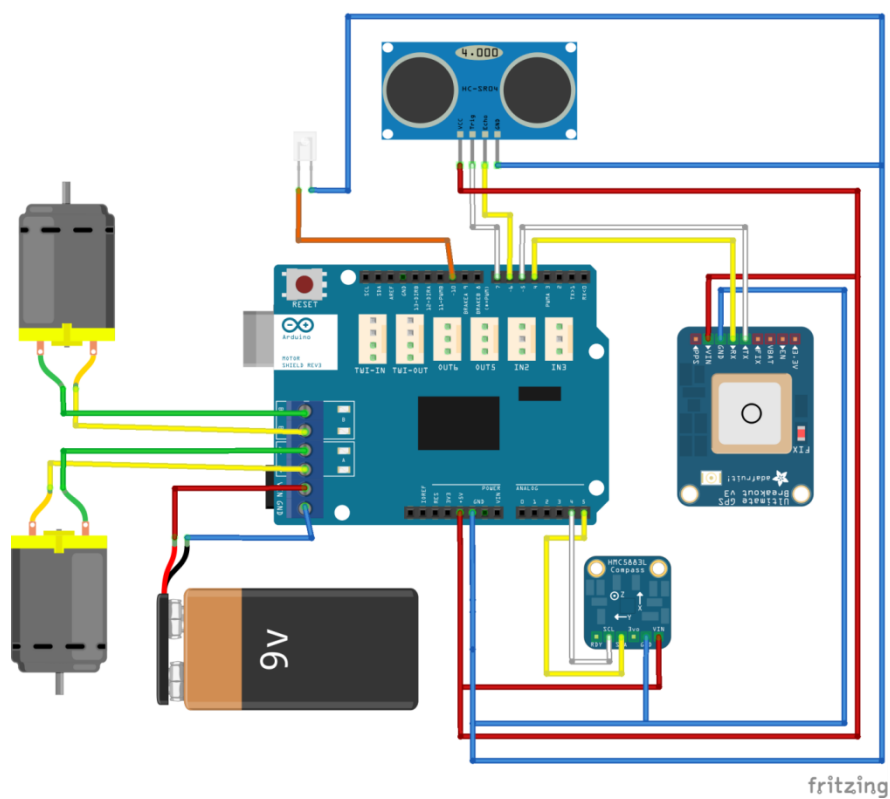
Konstrukce autonomního prostředku

6 Návrh řešení



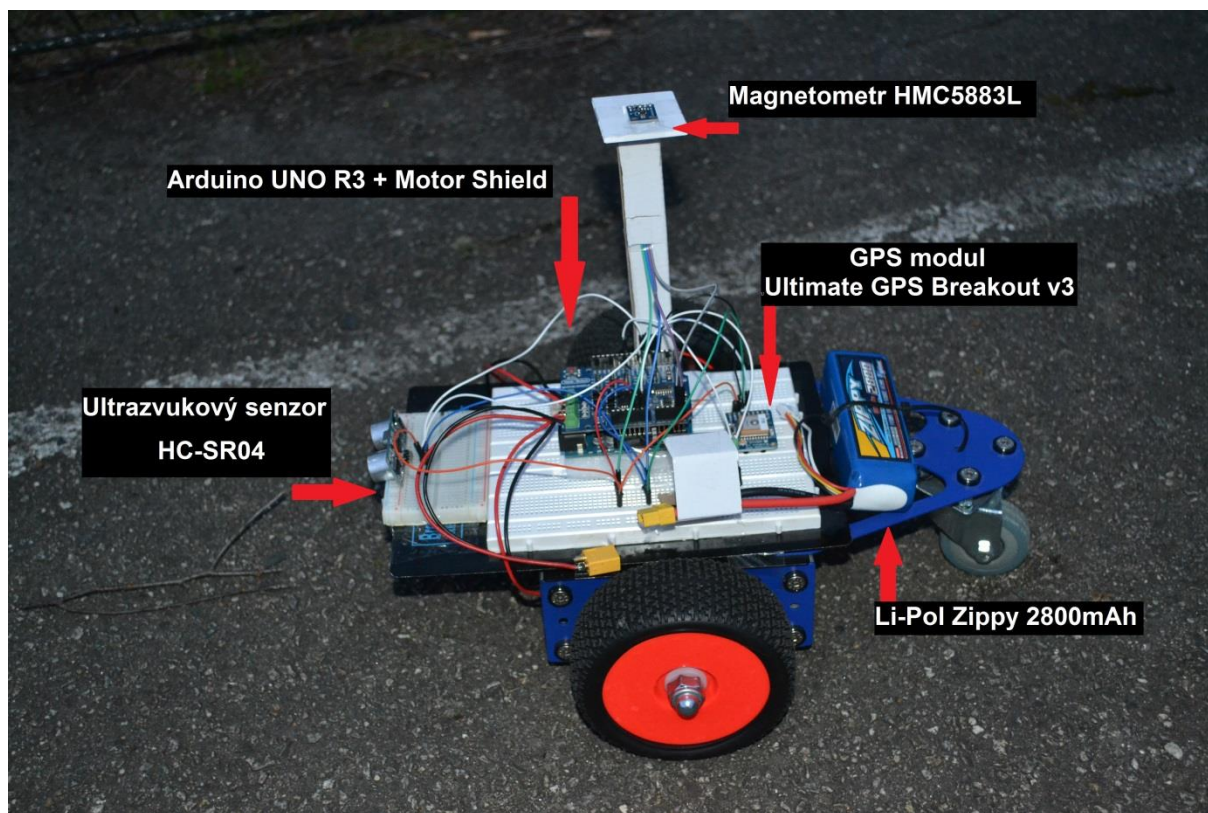
Obrázek 17: Blokové schéma

6.1 Zapojení prvků



Obrázek 18: Zapojení prvků

6.2 Rozdělení modulů



Obrázek 19: Rozdělení modulů

6.3 Algoritmizace

Hlavní část logiky je zobrazena ve zdrojovém kódu Zdrojový kód 2. Z GPS souřadnic a aktuálních souřadnic robota jsou zde vypočítány všechny důležité hodnoty pro další pohyb. Jako první se vypočítá vzdálenost do cíle. Ta se porovnává s hodnotou (přesností) lokalizace, která byla zjištěna při testování lokalizace. Dále je vypočítán kurz do cíle, který je porovnán se směrem do cíle. Podle těchto dvou hodnot se robot pohybuje směrem do cíle. Změna kurzu určuje, o kolik stupňů se robot otočí, aby směřoval do cíle.

6.4 Testování

6.4.1 Paměťová náročnost

Možnost pohybu, detekce překážek, navigace, lokalizace a přenosu dat vyžadují od řídicí jednotky velké množství úkonů a také obsluhu jednotlivých komponent. Pro jednodušší používání jsou od výrobců, nebo silné komunity uživatelů napsány knihovny. Tyto knihovny, zjednodušeně balíčky, jsou jakousi sbírkou kódů, které ulehčují programování senzorů a dalších připojitelných zařízení. Použitá knihovna v kódu se při nahrávání zkompiluje do paměti Arduina a společně s obslužným kódem vykonává svou činnost. Problém nastává v případě, kdy je použito velké množství knihoven a paměť je nedostačující.

Kód byl napsán ve dvou testovacích částech. První část kódu zajišťuje řídicí logiku robota, s tím spojené ovládání motorů. Je zde také ovládání GPS modulu spolu s elektromagnetickým kompasem. Souřadnice GPS jsou zde statické.

Potřebné knihovny (SoftwareSerial, TinyGPS++, Wire a HMC588L) spolu s obslužným kódem zabírají 50% (16 kB z 32 kB) celkové paměti.

Druhá část kódu se stará o komunikaci mezi webovou aplikací a robotem. Obsahuje knihovny (Adafruit_CC3000, ccspi, SPI, string, stdlib) a obslužný kód zabírají 73% (23 kB z 32 kB). Z těchto výsledků je patrné, že Arduino Uno není vhodné pro takto náročnou logiku a funkčnost.

Řešením by mohla být optimalizace knihoven a úspornější algoritmizace. Při testu optimalizace knihovny, se nepodařilo, uspořít očekávané místo. Navíc tato optimalizace probíhala velmi složitě. Použitím Arduina s větší paměťovou kapacitou by bylo možné tento problém řešit.

6.4.2 Detekce překážek

Po analýze algoritmů pro detekci překážek byl zvolen algoritmus Bug 0, který je jednoduchý na implementaci a zároveň nemusíme ukládat vzdálenost, směr a pohyb při

objíždění překážek. U jednoduchých překážek typu krabice byl algoritmus velmi efektivní. U speciálních ovšem není zaručena jeho spolehlivost.

6.4.3 Přesnost lokalizace

Při měření přesnosti lokalizace byl testován GPS modul Breakout v3. Pro ten byl napsán obslužný program, který zjišťoval aktuální GPS souřadnice a posílal je prostřednictvím wi-fi modulu do zhotovené webové aplikace. Ta tyto body vykreslovala do mapy. Na mapě byly zvoleny referenční body (například roh parkoviště) a na mapě podle měřítka změřena přesnost. Za jasného počasí nebyla naměřena větší odchylka než 3 metry. U mobilního prostředku, vzhledem k jeho konstrukci byla očekávána stejná přesnost.

7 Závěr

Při tvorbě bakalářské práce byly prostudovány možnosti pohybu a navigace pomocí řídicí jednotky Arduino Uno. Dále bylo ověřeno, zda je tato jednotka vhodná pro autonomní řízení vozidla. Ukázalo se, že Arduino Uno není příliš vhodné pro takto robustní zařízení, z důvodů omezeného počtu pinů a malé paměťové kapacity.

V teoretické části jsou nastíněny algoritmy pro detekci překážek a jejich základní rozdělení.

Při výběru čidel pro detekci překážek byly otestovány dva druhy snímačů, ultrazvuková a laser-optická čidla. Při konstrukci komunikačního mostu jsou nastíněny způsoby komunikace pomocí wi-fi.

Při vývoji robotu byly otestovány jednotlivé komponenty a popsány problémy při jejich konstrukci a spolupráci.

Autonomní pohyb a navigace má velký potenciál v návaznosti na autonomní drony, lodě apod. Do budoucího vývoje by mohl být algoritmus implementován i na tyto zařízení. Robot je díky použití GPS omezen na outdoorové prostředí. Navigace vně budovy by mohlo být předmětem dalšího zkoumání během navazujícího studia.

8 Literatura

- [1] TOCHÁČEK, Daniel a Jakub LAPEŠ. *Edukační robotika*. 1. vyd. Praha: Univerzita Karlova, Pedagogická fakulta, 2012, 52 s.
- [2] Jednodeskový počítač. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2015-04-22]. Dostupné z: http://cs.wikipedia.org/wiki/Jednodeskov%C3%BD_po%C4%8D%C3%ADta%C4%8D
- [3] ŠOLC, František. *Robotika, modelování a řízení robotů: Robotics, modelling and control of robots : teze přednášky ke jmenování profesorem v oboru "technická kybernetika"*. Brno: VUTIUM, 2004, 28 s.
- [4] SIEGWART, Roland. *Introduction to autonomous mobile robots*. 1. vyd. Massachusetts: MIT Press, 2004, 321 s.
- [5] THRUN, Sebastian. *Probabilistic robotics*. Massachusetts: MIT Press, c2006, xx, 647 s.
- [6] KLEČKA, Jan. *Pořízení a zpracování dat pro 2D a 3D SLAM úlohy robotické navigace* [online]. Brno, 2014, 67 l. [cit. 2015-04-23]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=82262. Diplomová práce. Vysoké učení technické v Brně.
- [7] Arduiniana. *TinyGPS++* [online]. © 2015 [cit. 2015-04-23]. Dostupné z: <http://arduiniana.org/libraries/tinygpsplus/>
- [8] HMC5883L: 3-Axis Digital Compass IC. *Adafruit Industries, Unique & fun DIY electronics and kits* [online]. b.r. [cit. 2016-04-21]. Dostupné z: https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf
- [9] KŮS, Václav, Jiří SKÁLA a Jiří HAMMERBAUER. *Elektromagnetická kompatibilita výkonových elektronických systémů*. 1. vyd. Praha: BEN - technická literatura, 2013, 372 s.
- [10] CHOSET, Howie M, Kevin LYNCH, Seth HUTCHINSON, George KANTOR, Wolfram BURGARD, Lydia KAVRAKI a Sebastian THRUN. *Principles of robot*

motion: theory, algorithms, and implementations. London, England: MIT Press, 2005, xix, 603 stran.

- [11] *Řízení autonomního vozidla pomocí umělé neuronové sítě*. Brno, 2005. Diplomová práce. Masarykova universita v Brně - Fakulta informatiky.
- [12] HC-SR04 User Guide: Ultrasonic Introduction. *Elecfreaks Store* [online]. b.r. [cit. 2016-04-21]. Dostupné z: http://elecfreaks.com/estore/download/EF03085-HC-SR04_Ultrasonic_Module_User_Guide.pdf
- [13] GP2Y0A02YK0F: Distance Measuring Sensor Unit. *SparkFun Electronics* [online]. b.r. [cit. 2016-04-21]. Dostupné z: https://www.sparkfun.com/datasheets/Sensors/Infrared/gp2y0a02yk_e.pdf
- [14] A RESTful environment for Arduino. *GitHub* [online]. 2016 [cit. 2016-04-21]. Dostupné z: <https://github.com/marcoschwartz/aREST>
- [15] PAUTASSO, Cesare, Erik WILDE a Rosa ALARCÓN. *REST: advanced research topics and practical applications*. New York: Springer, 2014, ix, 222 pages.
- [16] STEINER, Ivo a Jiří ČERNÝ. *GPS od A do Z*. 4., aktualiz. vyd. Praha: eNav, 2006, 264 s.
- [17] HOJGR, Radek a Jan STANKOVIČ. *GPS: praktická uživatelská příručka*. Vyd. 1. Brno: Computer Press, 2007, 221 s.
- [18] Gpsinformation. *NMEA data* [online]. 2013 [cit. 2015-04-23]. Dostupné z: <http://www.gpsinformation.org/dale/nmea.htm>
- [19] Vestavěný systém. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2015-04-22]. Dostupné z: http://cs.wikipedia.org/wiki/Vestav%C4%9Bn%C3%BD_syst%C3%A9m

9 Přílohy

9.1 Seznam Obrázků

Obrázek 1: Blokové schéma robota [3]	14
Obrázek 2: Schematické znázornění SLAM algoritmu [6]	17
Obrázek 3: Arduino UNO R3	18
Obrázek 4: Integrovaný obvod L293D	20
Obrázek 5: Arduino Motor Shield R3.....	21
Obrázek 6: Arduino Wi-Fi Shield.....	22
Obrázek 7: GPS modul Ultimate Breakout v3.....	24
Obrázek 8: Blokové schéma výpočtu kurzu z pohybu prostředku	26
Obrázek 9: Trojosý magnetometr HMC5883L.....	27
Obrázek 10: Algoritmus Bug 0	31
Obrázek 11: Algoritmus Bug 1	31
Obrázek 12: Algoritmus Bug 2	32
Obrázek 13: Neuronová síť pro vyhýbání se překážkám	33
Obrázek 14: Ultrazvukový senzor HC-SR04 [12].....	34
Obrázek 15: Čidlo vzdálenosti Sharp 2Y0A02.....	35
Obrázek 17: Webová aplikace	42
Obrázek 17: Blokové schéma	43
Obrázek 18: Zapojení prvků	43
Obrázek 19: Rozdělení modulů	44

9.2 Seznam Tabulek

Tabulka 1: Motor Shield, zapojení pinů k Arduino	21
Tabulka 2: Zapojení Wi-Fi shieldu	23
Tabulka 3: Popis RMC věty [18]	38
Tabulka 4: Popis GGA věty [18]	39
Tabulka 5: Popis GSA věty [18].....	40
Tabulka 6: Popis GSV věty [18].....	40

9.3 Seznam Zdrojových kódů

Zdrojový kód 1: Výpočet kurzu v knihovně TinyGPS++ [7]..... 25

Zdrojový kód 2: Algoritmus pro výpočet a pohyb robota.....45

9.4 Struktura přiloženého CD

\library	použité knihovny
\source	zdrojové kódy k jednotlivým programům
\text	bakalářská práce ve formátu PDF