

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

**NÁVRH A REALIZACE KOMUNIKAČNÍHO PROTOKOLU A
SOFTWARE PRO BEZDRÁTOVÝ PŘENOS DAT MEZI
ROBOTICKÝM VOZÍTKEM A MATLABEM**

Vojtěch Ihnát

Bakalářská práce
2016

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Vojtěch Ihnát**
Osobní číslo: **I12066**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Řízení procesů**
Název tématu: **Návrh a realizace komunikačního protokolu a softwaru pro bezdrátový přenos dat mezi robotickým vozítkem a MATLABem**
Zadávající katedra: **Katedra řízení procesů**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je návrh a realizace komunikačního protokolu pro obousměrný bezdrátový přenos dat mezi robotickými vozítky a MATLABem, kde MATLAB představuje nadřazený systém zajišťující návrh trajektorie pohybu robotů. Protokol musí umožnit současné nasazení až pěti robotů. Komunikační protokol musí být navržen tak, aby zajišťoval kompatibilitu jak s řídicím software robota, tak i plánovacím systémem na straně MATLABu. Součástí práce je jak verifikace vytvořeného softwaru, tak i ověření dosahu a spolehlivosti zvolné technologie přenosu dat.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

**MATLAB Documentation - MathWorks [online]. c2014 [cit. 2014-10-08]. . HC
Serial Bluetooth Products - User Instructional Manual [online]. c2014 [cit.
2014-10-09].**

Arduino UNO Reference Design [online]. c2014 [cit. 2014-10-09].

Vedoucí bakalářské práce:

Ing. Pavel Škrabánek, Ph.D.

Katedra řízení procesů

Datum zadání bakalářské práce:

4. prosince 2014

Termín odevzdání bakalářské práce:

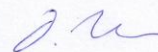
11. května 2015



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2015

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 02. 05. 2016

Vojtěch Ihnát

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu panu Ing. Pavlu Škrabánkovi Ph.D, za veškerou pomoc a ochotu při tvorbě této bakalářské práce. Také bych chtěl poděkovat své rodině za podporu během studia.

V Pardubicích dne 02. 05. 2016

Vojtěch Ihnát

ANOTACE

Cílem této práce je návrh a realizace komunikačního protokolu mezi prostředím MATLAB a robotem. Komunikační protokol musí umožnit současné nasazení až pěti robotů. MATLAB představuje řídicí systém.

KLÍČOVÁ SLOVA

MATLAB, Arduino Uno, bluetooth, robot, HC-05.

TITLE

COMMUNICATION PROTOCOL DESIGN AND SOFTWARE DEVELOPMENT FOR THE WIRELESS DATA TRANSMISSION AMONG MOBILE ROBOTS AND MATLAB

ANNOTATION

The aim of this work is the design and implementation of communication protocol between MATLAB and the robot. Communication protocol must allow current use of up to five robots. MATLAB will be control system.

KEYWORDS

MATLAB, Arduino Uno, Bluetooth, Robot, HC-05.

Obsah

Seznam zkratek	8
Seznam značek	9
Seznam obrázků	10
Seznam tabulek	11
ÚVOD	12
1 POPIS LABORATORNÍHO MODELU	13
1.1 BLUDIŠTĚ	13
1.2 KAMEROVÝ SYSTÉM.....	14
1.3 ROBOT	15
2 PLÁNOVÁNÍ CESTY	16
2.1 PŘEHLED ALGORITMŮ PRO PLÁNOVÁNÍ CESTY.....	16
2.1.1 Algoritmus roadmap	16
2.1.2 Algoritmus cell decomposition	16
2.1.3 Algoritmus potential field	17
2.2 ZPRACOVÁNÍ NALEZENÉ CESTY	17
3 KOMUNIKACE	19
3.1 KOMUNIKAČNÍ PROTOKOL	19
3.2 NÁVRH A REALIZACE KOMUNIKAČNÍHO PROTOKOLU	19
3.3 PŘENOS DAT	21
3.3.1 Způsoby přenosu dat	21
3.4 BEZDRÁTOVÁ KOMUNIKACE	24
3.4.1 Bezdrátová lokální síť (WLAN)	24
3.4.2 Bezdrátová osobní síť (WPAN).....	25
3.4.3 Bezdrátová rozsáhlá síť (WWAN).....	26
3.5 BLUETOOTH TECHNOLOGIE	26
3.5.1 Standardy	27
3.5.2 Bluetooth architektura.....	27
3.5.3 Bluetooth modul HC-05.....	29
4 PRAKTICKÁ REALIZACE.....	32
4.1 MATLAB MATHWORKS.....	32
4.2 ARDUINO	34
4.2.1 Arduino Uno	34

4.2.2	Vývojové prostředí Arduina.....	35
4.3	PRAKTICKÉ ŘEŠENÍ	37
4.3.1	Připojení modulů HC-05 k Arduino Uno platformě	37
4.3.2	Spárování zařízení.....	38
4.3.3	Nastavení v Arduinu	40
4.3.4	Nastavení v MATLABu.....	41
5	EXPERIMENTÁLNÍ OVĚŘENÍ	43
6	ZÁVĚR	45
	LITERATURA.....	46
	Seznam příloh	47

Seznam zkratek

ISM	rádiové pásmo (industrial, scientific and medical)
IDE	integrované vývojové prostředí
MATLAB	Matrix Laboratory
PDA	osobní asistent
WPAN	osobní bezdrátová síť
WLAN	bezdrátová lokální síť
WWAN	bezdrátová rozsáhlá síť

Seznam značek

D	klopný obvod D
G	graf
E	hrana grafu
e	hrany grafu
m	počet stavů číslicového signálu
V	vrchol grafu
v	vrcholy grafu
Q	sériový výstup
T	hodinový signál
v_m	modulační rychlost, Bd
v_p	přenosová rychlost, Bd
x	osa x
y	osa y

Seznam obrázků

Obr. 1.1 – Bludiště	14
Obr. 1.2 – Kamerový systém	14
Obr. 1.3 – Směr analýzy bludiště.....	15
Obr. 2.1 – Algoritmus potential field.....	17
Obr. 2.2 – Naplánovaná cesta	18
Obr. 2.3 – Seznam příkazů.....	18
Obr. 3.1 – Demonstrovaný problém s přenosem dat	20
Obr. 3.2 – Vytvořený komunikační protokol.....	20
Obr. 3.3 – Posuvný registr	21
Obr. 3.4 – Příklad modulační rychlosti.....	22
Obr. 3.5 – Způsoby přenosu dat.....	23
Obr. 3.6 – Router	24
Obr. 3.7 – Síť WLAN	25
Obr. 3.8 – Síť WPAN	25
Obr. 3.9 – Síť WWAN.....	26
Obr. 3.10 – Logo bluetooth.....	26
Obr. 3.11 – Bluetooth protokoly a vrstvy	28
Obr. 3.12 – Spojení piconet a scatternet	30
Obr. 4.1 – Logo MATLAB	32
Obr. 4.2 – Grafické prostředí MATLAB R2012a.....	33
Obr. 4.3 – Platforma Arduino Uno	34
Obr. 4.4 – Vývojové prostředí Arduina	36
Obr. 4.5 – Sériový monitor Arduina	36
Obr. 4.6 – Bluetooth modul HC-05	37
Obr. 4.7 – Připojení bluetooth modulu HC-05 k Arduino Uno platformě.....	38
Obr. 4.8 – Hlavní nabídka.....	38
Obr. 4.9 – Přidat zařízení bluetooth.....	39
Obr. 4.10 – Zadání párovacího kódu bluetooth	39
Obr. 4.11 – Schéma zapojení pro uvedení modulu HC-05 do příkazového režimu	41

Seznam tabulek

Tab. 3.1 – Třídy bluetooth technologie	29
Tab. 3.2 – Přenosové rychlosti.....	30
Tab. 3.3 – Tabulka AT příkazů.....	31
Tab. 5.1 – Tabulka spolehlivosti.....	44

ÚVOD

Cílem této práce je navrhnout komunikační protokol pro bezdrátový přenos dat mezi MATLABem a robotickými vozítky i následná verifikace vytvořeného softwaru. MATLAB v tomto případě představuje nadřazený systém, který zajišťuje plánování pohybu robotů na základě dat přijatých z kamerového systému a požadavků uživatele. Nadřazený systém i robotická vozítka jsou součástí laboratorní úlohy určené pro výuku plánovacích metod v rámci předmětu „Základy umělé inteligence I“.

Tato práce je rozdělená do šesti kapitol. První kapitola se zabývá laboratorní úlohou. V této kapitole je popsáno bludiště, které tvoří pracovní prostor pro roboty. Dále je zde popsán kamerový systém, který slouží k analýze struktury bludiště a určení polohy robotů. V poslední podkapitole jsou základní informace o robotovi.

Druhá kapitola se zabývá plánováním cesty, následným zpracováním nalezené cesty, a také je zde uveden přehled algoritmů pro plánování cesty.

Třetí kapitola je věnována komunikačnímu protokolu. Pro komunikaci mezi nadřazeným systémem a roboty, je využíváno bezdrátové komunikační technologie bluetooth. V této kapitole se nachází nezbytná teorie, také popis řešení, včetně analýzy problému.

Čtvrtá kapitola popisuje implementaci navrženého řešení. Je zde popsán použitý software a hardware.

V páté kapitole je popsán způsob ověření funkčnosti navrženého řešení, včetně testu jeho dosahu a spolehlivosti.

V poslední kapitole jsou shrnuty a diskutovány výsledky této práce.

1 POPIS LABORATORNÍHO MODELU

V této části je popsáno bludiště a kamerový systém, které máme k dispozici. Bludiště tvoří pracovní prostor pro roboty. Kamerový systém slouží k analýze struktury bludiště a určení polohy robotů. Dále jsou zde uvedeny jen základní informace o robotovi, protože zpracováním dat a sestavením robota se zabýval kolega Pavel Vodička (Vodička, 2016).

1.1 BLUDIŠTĚ

Základem bludiště je podlaha bílé barvy. Struktura bludiště je sestavena z příček a pilířů. Velikost příček je vždy stejná, jejich délka je 17 cm, šířka 1 cm a výška 10 cm, pilíře o výšce 10 cm a čtvercové podstavy 1 cm x 1 cm jsou pevně připevněny k podlaze.

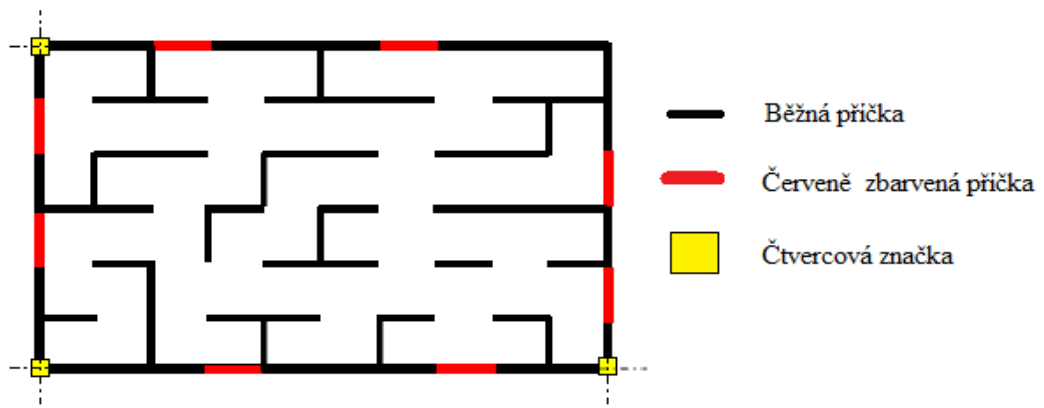
Příčky tvoří pravouhlý systém, tím vznikne mřížka obr. 1.3, jak je z tohoto obrázku patrné, mřížky neboli buňky mají vždy stejnou velikost.

Příčky, které jsou umístěny na vnějším okraji bludiště, jsou bílé barvy, kromě jednoho okraje. Tyto okraje jsou u většiny příček černé barvy, s výjimkou osmi červených.

Příčky jsou připevněny k podlaze pomocí pilířů. Na kolmých stranách pilíře se vyskytuje drážka, do kterých se jednotlivé příčky vkládají, to znamená, že můžeme bludiště modifikovat podle potřeb. Avšak musí být zachováno několik podmínek. První podmínkou je, že vnější tvar bludiště musí být obdélníkový, dále se musí jednat o uzavřený systém (izolovaný od okolního světa) a rozměry celého bludiště by měli respektovat omezení kamerového systému.

Příčky musí být vloženy do pilířů tak, aby byl vidět barevný okraj při pohledu ze shora. Červeně zbarvené okraje příček jsou nezbytné k analýze bludiště a každá vnější strana musí obsahovat alespoň jednu nebo dvě, takhle zbarvené příčky a minimálně čtyři by měli být použity v celém bludišti, abychom zajistili správnou analýzu dat. Jejich umístění musí splňovat dva požadavky. Prvním požadavkem je, že nesmí být dvě červené příčky umístěny hned vedle sebe a dále nesmí být umístěny hned vedle rohu bludiště. Avšak jejich rozložení by mělo být vyvážené, abychom minimalizovali chyby způsobené umístěním kamery.

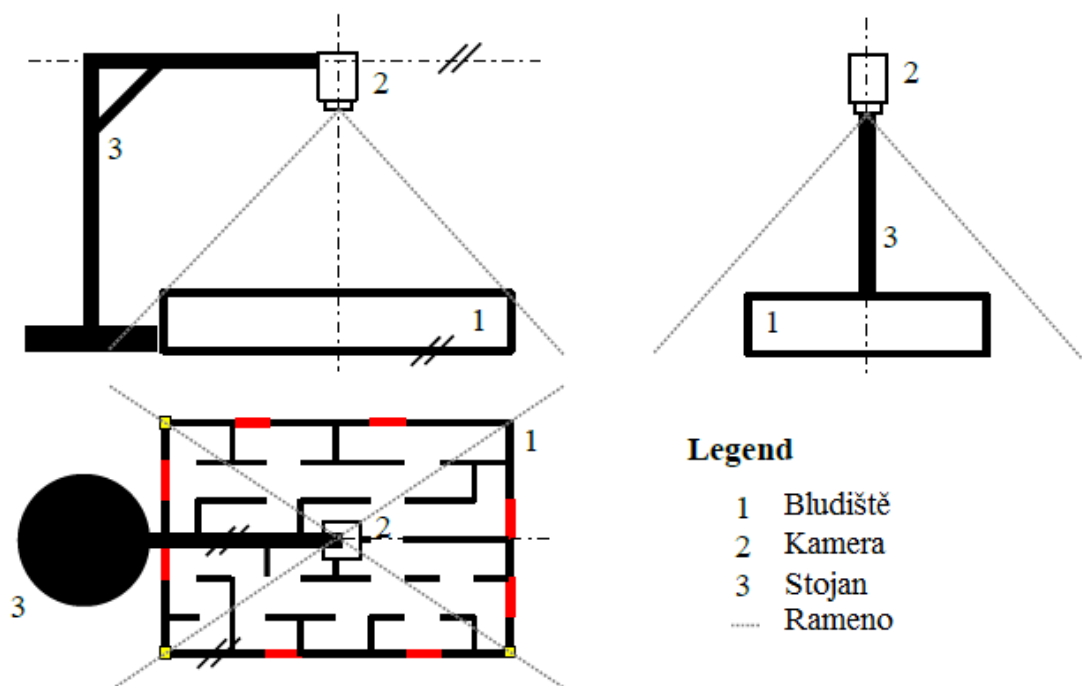
Jakmile je bludiště sestaveno, umístíme tři žluté čtvercové značky do rohů bludiště. Dvě značky budou umístěny v dolních rozích a jeden v levém horním rohu, jak je názorně ukázáno na obr. 1.1 (Škrabánek, 2015b).



Obr. 1.1 – Bludiště, (Škrabánek, 2015b)

1.2 KAMEROVÝ SYSTÉM

Kamerový systém se skládá z podstavce, stojanu a kamery. Kamera je připevněna na konci stojanu a její objektiv je umístěn ve středu bludiště. Rameno stojanu by mělo být vodorovně s podlahou a okraje obrázků by měli být vodorovné s vnějšími stěnami bludiště. Výška by měla být zvolena tak, aby bylo v záběru celé bludiště a minimum okolí. Příklad správného umístění kamery je na obr. 1.2 (Škrabánek, 2015b).



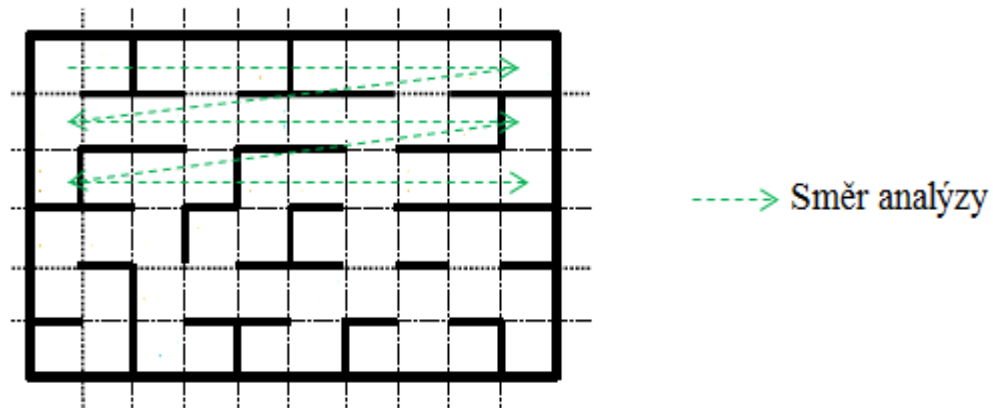
Obr. 1.2 – Kamerový systém, (Škrabánek, 2015b)

Jak již bylo výše zmíněno, kamerový systém nám slouží k analýze bludiště. Avšak musí být provedena dekompozice z reálného světa do vhodné diskrétní reprezentace.

Vzhledem k rozměrům bludiště a příček, velikost jednotlivých buněk je vždy stejná a mají čtvercový tvar, to znamená, že tvoří mřížku, jak je patrné z obr. 1.3. Proto nejlepší

metodou pro dekompozici bludiště je metoda přesného rozložení do buněk (*exact cell decomposition*). Bližší informace o této metodě jsou uvedeny v kapitole 2.

Analýza bludiště probíhá ve smyčce a postupuje z levé strany na pravou, řádek po řádku, jak je to znázorněno na obr. 1.3. K analýze se nepoužívá přímo obrázek z kamery, nejdříve se musí transformovat na binární obraz.



Obr. 1.3 – Směr analýzy bludiště

Kamerový systém umí určit i polohu a orientaci robota, tyto informace se dále pak používají v plánovacím algoritmu.

1.3 ROBOT

Robot je autonomní. Posílají se mu příkazy na základě dat přijatých z kamerového systému, které jsou následně vyhodnocovány plánovacím algoritmem. Lokalizace robota probíhá pomocí dat přijatých ze senzorů.

Robot je vybaven dvěma předními a dvěma bočními senzory, inkrementálními optickými snímači a tříosovým magnetometrem. Také má diferenciální podvozek, to znamená, že má dvě poháněná kola. Každé z nich je poháněno motorem L293D.

Detekce překážek je realizována pomocí výše zmíněných senzorů a magnetometr slouží k detekci magnetických značek, které jsou umístěny na povrchu bludiště. Tyto značky umožňují robotovi jeho lokalizaci.

Řízení celého robota, včetně zpracování dat a komunikace je zajištěno pomocí platformy Arduino Uno. Komunikace s jinými zařízeními probíhá pomocí USB kabelu nebo pomocí bluetooth modulu HC-05. Bližší informace o platformě Arduino Uno a bluetooth modulu HC-05 jsou v následujících kapitolách (HC-05, 2010).

2 PLÁNOVÁNÍ CESTY

Cílem je najít nejkratší cestu mezi počátečním stavem a cílovým, nicméně cesta by se měla vyhnout jakékoliv kolizi s některou dopředu známou překážkou v pracovní oblasti robota. K nalezení této cesty slouží plánovací algoritmy, které jsou zmíněné níže.

Avšak nejdříve se musí provést analýza bludiště. Jak již bylo zmíněno v předchozí kapitole, analýza bludiště probíhá pomocí kamerového systému. Výsledkem analýzy je graf G , kde každá buňka představuje vrchol grafu V a možnost přechodu mezi dvěma sousedícími buňkami je hranou grafu E .

$$G(V, E), \tag{2.1}$$

kde $V = \{v_1, v_2, \dots\}$ – vrcholy grafu,

$E = \{e_1, e_2, \dots\}$ – hrany grafu.

Podrobný popis plánování cesty je popsán v literatuře (Škrabánek, 2015a).

2.1 PŘEHLED ALGORITMŮ PRO PLÁNOVÁNÍ CESTY

Existuje mnoho metod a algoritmů pro plánování cesty. Některé z nich umožňují detekovat překážky, které jsou definovány jako mnohoúhelníky například graf viditelnosti. Podrobný popis těchto algoritmů je uveden v literatuře (Siegwart, 2011). Níže je uvedeno základní členění:

- 1) mapy cest (*roadmap*),
- 2) rozklad do buněk (*cell decomposition*),
- 3) potencionální pole (*potential field*).

2.1.1 Algoritmus roadmap

Tato metoda rozloží volný prostor do sítě 1-D křivky nebo řádků, čímž získáme graf, který znázorňuje daný problém. Řešením bývá nejkratší cesta.

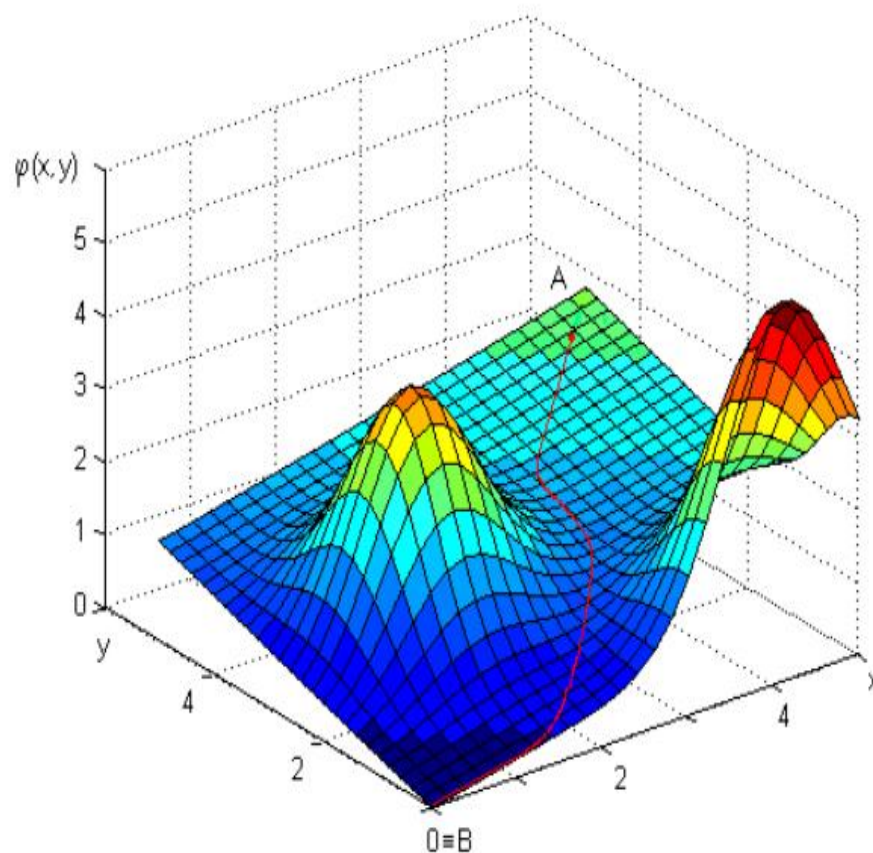
2.1.2 Algoritmus cell decomposition

Zatímco tato metoda rozloží volný prostor na sadu jednoduchých buněk, takže je možné určit, zda je buňka prázdná nebo jestli obsahuje překážku. Tato metoda může být ještě rozdělená na přesné rozložení buněk (*exact cell decomposition*) a na přibližné rozdělení buněk

(*approximatecell decomposition*) vzhledem k hranicím mezi buňkami. U metody *approximatecell decomposition* se rozkládá do buněk celý prostor včetně překážek, proto může nastat problém, že tato metoda nenajde cestu, i když existuje. Kdežto metoda *exact cell decomposition* rozdělí prostor do vzájemně nepřekrývajících se buněk. Tato metoda vždy nalezne cestu, pokud tedy existuje, v opačném případě ověří, že neexistuje.

2.1.3 Algoritmus potential field

Celý pracovní prostor je pokryt potenciálním polem (mřížkou). Cílová pozice má menší potenciál než počáteční, zatímco místa, kde jsou překážky, působí jako odpudivé síly (tvoří „vrcholky“, (Šíma, 2000).



Obr. 2.1 – Algoritmus potential field (Šíma, 2000).

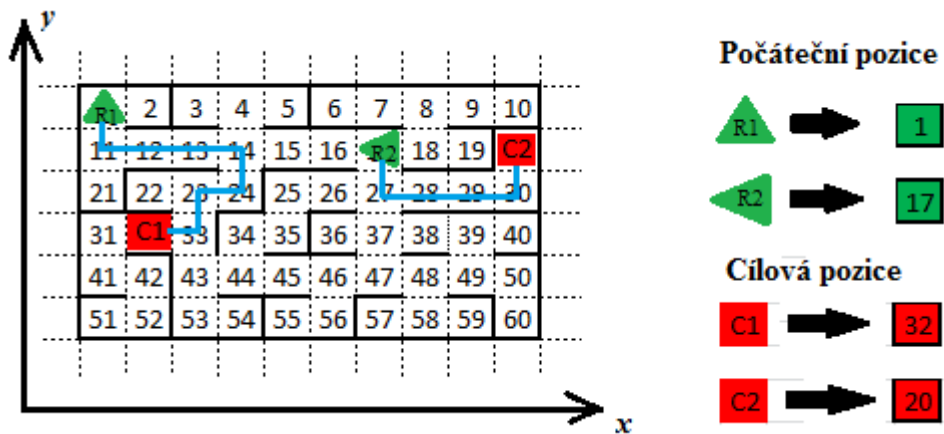
2.2 ZPRACOVÁNÍ NALEZENÉ CESTY

Plánovací algoritmus nejdříve nalezne cestu z počáteční pozice robota do cílové. Poté se určí, které buňky bludiště má robot projet, jak je patrné z obr. 2.2, kdy se robot R2 má dostat do cílové buňky 20. Z toho nám vznikne posloupnost buněk. Tato posloupnost se dále převede na sekvenci příkazů, kde každé číslo reprezentuje určitou akci, která se má vykonat.

Aby se Robot R2 dostal do své sílové buňky musí vykonat tyto akce:

1. Otočit se doleva, tedy o -90°
2. Dostane se do buňky 27 a znovu se otočí o -90°
3. Poté jede rovně až do buňky 30, kde se zastaví
4. V buňce 30 se naposled otočí o 90° a tím se dostane do své cílové buňky 20.

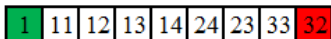
Posloupnost buněk a příkazů je zobrazena na obr. 2.3.



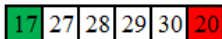
Obr. 2.2 – Naplánovaná cesta

Posloupnost buněk

Robot R1

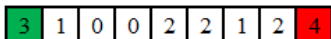


Robot R2

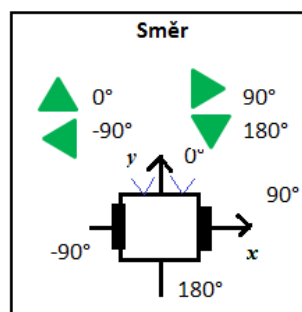
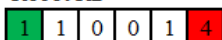


Posloupnost příkazů

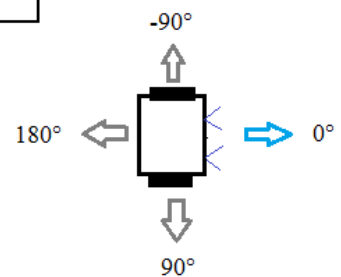
Robot R1



Robot R2



Příkaz	Akce
0	0°
1	-90°
2	90°
3	180°
4	STOP



Obr. 2.3 – Seznam příkazů

3 KOMUNIKACE

V této kapitole je čtenář seznámen s pojmem komunikační protokol a také je zde uveden návrh a realizace vlastního komunikačního protokolu. Dále jsou zde popsány možné způsoby přenosu dat, bezdrátová komunikace a bluetooth modul HC-05, který byl použit v praktické realizaci pro bezdrátový přenos dat.

3.1 KOMUNIKAČNÍ PROTOKOL

Komunikační protokol definuje postupy a parametry, které se používají při vysílání a příjmu dat, jako je například definice formátů a použitých identifikátorů.

V tomto případě je zapotřebí vyhodnotit data z kamerového systému v MATLABu, přiřadit jim identifikátor, který zajistí, aby se data neopakovaly a následně odeslat tyto data do robotických vozíků.

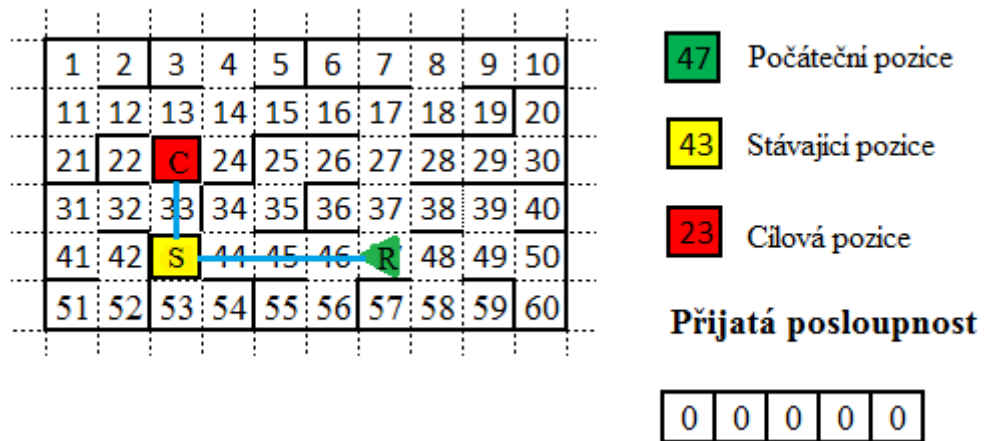
3.2 NÁVRH A REALIZACE KOMUNIKAČNÍHO PROTOKOLU

Jak již bylo zmíněno v kapitole 2, plánovací algoritmus vygeneruje cestu na základě polohy robota a požadavků uživatele. Tím vznikne posloupnost buněk, které se mají projet. Posloupnost buněk může mít rozdílnou velikost, odvíjí se to od celkové vzdálenosti z počátečního bodu do cílového bodu. Jak je to například znázorněno na obr. 2.2 a 2.3, kdy robot R1 musí ujet delší vzdálenost do svého cíle než robot R2.

Tyto informace se dále rozdělí na posloupnost pěti příkazů, jak je to blíže popsáno v kapitole 2.2. Posloupnost pěti příkazů byla zvolena záměrně, protože může nastat problém s bezdrátovým přenosem dat. To znamená, že jakmile se naváže komunikace, pošle se tato posloupnost. V případě, že se komunikace přeruší, robot uchovává ve své paměti tuto posloupnost a může postupně vykonávat jednotlivé příkazy. Mezitím může nastat situace, kdy se opět podařilo navázat komunikaci a robot obdrží další posloupnost.

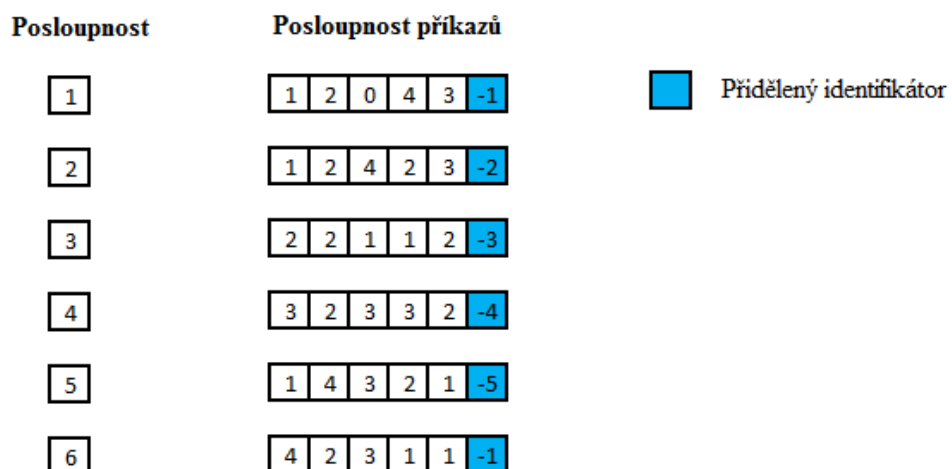
Nicméně může nastat další problém, kdy se příchozí posloupnost shoduje s předchozí. Tato situace je demonstrována na obr. 3.1. Kdy se robot R snaží dostat ze své počáteční pozice, v tomto případě buňky 47, do své cílové pozice buňka 23. Robot obdržel na začátku komunikace posloupnost pěti příkazů, v tomto případě samé nuly, což znamená, že pojedou rovně. Nicméně v průběhu vykonávání těchto pěti příkazů nastal problém a bezdrátová komunikace se přerušila. Mezitím robot vykonal veškeré příkazy, které měl uchované v paměti a zastavil se v buňce 43. Kdyby v této situaci došlo k opětovnému navázání

komunikace a robot R by obdržel stejnou posloupnost, jako předchozí mohlo by dojít ke kolizi, protože se před ním nachází překážka. Z toho důvodu se přidělí k jednotlivým posloupnostem tzv. identifikátor, tím se zajistí, aby se posloupnosti neopakovaly.



Obr. 3.1 – Demonstrováný problém s přenosem dat

Přidáním identifikátoru tedy vznikne posloupnost šesti příkazů. Jako identifikátor byla zvolena celá záporná čísla od -1 až do -5. Tato čísla byla zvolena z toho důvodu, že ke splnění naplánované cesty, by mělo stačit pět sekvencí příkazů. V případě, že naplánovaná cesty je dlouhá a je zapotřebí více sekvencí, než je identifikátorů, tak se začnou přiřazovat znovu od -1. Příklad přiřazených identifikátorů je na obr. 3.2



Obr. 3.2 – Vytvořený komunikační protokol

3.3 PŘENOS DAT

Přenos dat může probíhat mezi dvěma zařízeními nebo mezi více zařízeními. Nejdříve však musí dojít k inicializaci. To znamená, že zařízení, které chce data vysílat, oznámí tuto skutečnost zařízením, které budou data přijímat. Dále dochází k adresování, každé zařízení má svojí vlastní adresu její délka bývá obvykle 4 a 8 bitové číslo. Tato adresa bývá uložena v paměti, v některých případech je i nastavitelná pomocí přepínačů.

Vysílač tuto adresu odešle na začátku přenosu a zařízení, které tuto adresu vlastní se připraví k příjmu. Ostatní zařízení jsou v klidu. Poté dochází k přenosu vlastní zprávy, kde začátek přenosu je označen start bitem a konec je označen stop bitem. Následně přijímač potvrdí přijetí zprávy.

Způsob přenosu dat se dělí na sériový, paralelní, synchronní a asynchronní.

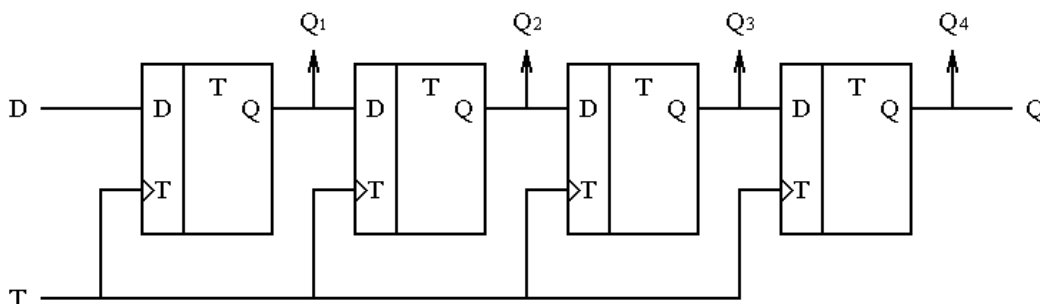
3.3.1 Způsoby přenosu dat

1) **Sériový** – data, která se mají odeslat, se vysílají v časové posloupnosti (sekvenčně) a pro přenos se využívá jeden přenosový kanál nebo sběrnice. Tedy data se posílají bit po bitu. Sériový přenos se využívá pro přenos na větší vzdálenosti.

2) **Paralelní** – pro každý prvek se používá samostatný přenosový kanál. Tedy přenáší se celé slovo současně, které má obvykle 8 nebo 16 bitů. Slouží pro přenos na krátké vzdálenosti pro vysokorychlostní přenosy dat.

Sériový přenos je úspornější z hlediska počtu vodičů a zaručuje větší spolehlivost. Avšak data, které se mají odeslat, jsou obvykle v paralelním tvaru. Před přenosem se tedy musí převést na sériový tvar, po přenosu se převedou zase na paralelní tvar, k tomu nám slouží posuvný registr.

Posuvný registr je skupina klopných obvodů, které mají propojené vstupy a výstupy.



Obr. 3.3 – Posuvný registr

3) **Synchronní** – dva za sebou následující charakteristické okamžiky jsou násobkem jednotkového intervalu. Mají tedy společný hodinový signál.

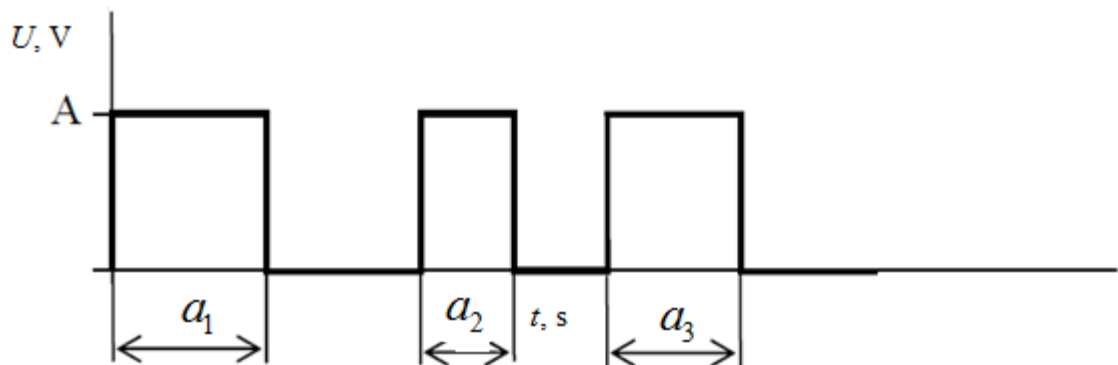
4) **Asynchronní** – asynchronní neboli arytmičtý přenos nemá společný hodinový signál pro přijímač i vysílač, který by stanovil interval platnosti značek (např. data - byty). Proto se musí přenášet informace o začátku a konci prvku. Například bitovou posloupností nebo start bitem a stop bitem (Vlček).

U těchto přenosů se rozlišuje například modulační rychlost a přenosová rychlost.

Modulační rychlost je vlastně počet symbolů přenesených za sekundu. Je to převrácená hodnota délky nejkratšího nezkresleného signálu. Udává se v jednotkách Baud, zkratka Bd a rozměr této veličiny je 1/s.

$$v_m = \frac{1}{\min(a)}, \quad (3.1)$$

kde v_m – modulační rychlost,
 a – nejkratší nezkreslený signál.



Obr. 3.4 – Příklad modulační rychlosti

Přenosová rychlost nám udává, kolik bitů je možné přenést za jednotku času.

Jednotky jsou v bitech za sekundu.

$$v_p = v_m \cdot \log_2 m, \quad (3.2)$$

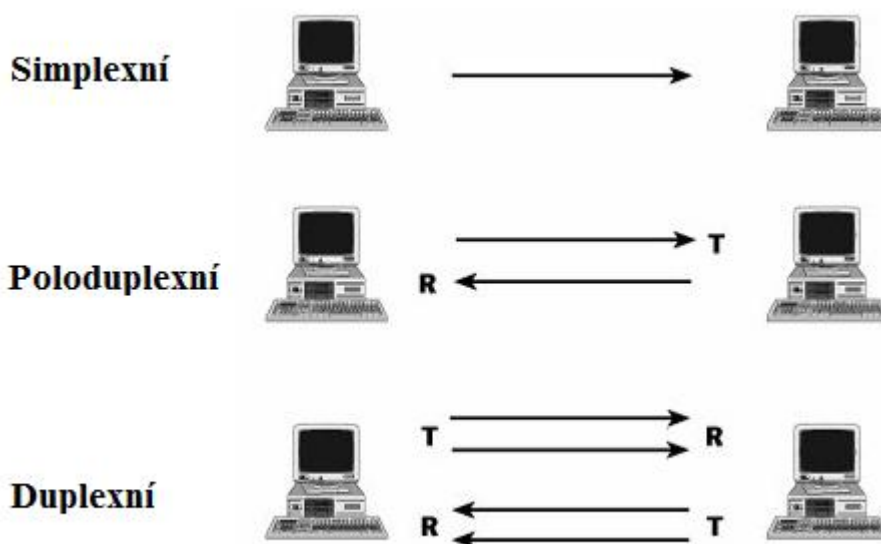
kde v_p – přenosová rychlost,
 v_m – modulační rychlost,
 m – počet stavů číslicového signálu.

Dále se rozlišuje způsob spojení neboli komunikace. Ty se dělí následovně: simplexní, poloduplexní a duplexní obr. 3.5.

1) **Simplexní** – jedná se o způsob komunikace, kdy přenos informací je možný pouze v jednom směru. Tedy buď může jen přijímat nebo odesílat, používá se pouze jeden přidělený kanál.

2) **Poloduplexní** – v tomto případě opět stačí pouze jeden přidělený kanál. Komunikace může probíhat obousměrně, ale ne současně. Jednotlivé směry přenosu se musí přepínat.

3) **Duplexní** – Komunikace probíhá v obou směrech současně, je tedy zapotřebí dvou komunikačních kanálů.



Obr. 3.5 – Způsoby přenosu dat

3.4 BEZDRÁTOVÁ KOMUNIKACE

Bezdrátová komunikace se využívá v mnoha odvětvích. Umožňuje spojení na krátké vzdálenosti (např. bluetooth, Wi-fi) až do milionů kilometrů (komunikace družic).

Mezi zařízení, která se používají běžně pro bezdrátový přenos, patří například: stolní počítač nebo mobilní telefon. Uživatelé se mohou připojit k síti internet pomocí bezdrátové komunikace nebo si mohou doma synchronizovat data mezi zařízeními a přenášet je dle libosti.

Rozlišujeme tři základní druhy bezdrátových sítí. Popsány jsou níže.

3.4.1 Bezdrátová lokální síť (WLAN)

(Wireless Local Area Network) Spojuje dvě a více zařízení v omezeném prostoru, například: doma, v kanceláři a ve škole. K přenosu slouží rádiové vlny v řádech GHz (gigahertzů). Umožňuje připojení k internetu pomocí bezdrátového směrovače tzv. routeru obr. 3.6. Předpokládány dosah signálu je 45 až 90 metrů, závisí to na fyzickém prostředí.



Obr. 3.6 – Router (Žákavcová, 2009)



Obr. 3.7 – Síť WLAN (Žákavcová, 2009)

3.4.2 Bezdrátová osobní síť (WPAN)

(Wireless Personal Area Network) Tato síť umožňuje propojit zařízení mezi sebou, například mobilní telefony pomocí bluetooth technologie. Pracuje na krátké vzdálenosti, předpokládaný dosah je 10 metrů.



Obr. 3.8 – Síť WPAN (Žákavcová, 2009)

3.4.3 Bezdrátová rozsáhlá síť (WWAN)

(Wireless Wide Area Network) Již z názvu vyplývá, že se jedná o síť, která pokrývá rozsáhle oblasti například mezi vesnicemi a městy. Tuto síť obvykle poskytují mobilní operátoři. V dnešní době se nejčastěji využívá mobilní síť GSM pod kterou spadají služby GPRS a EDGE.

Tato síť je označována jako síť druhé generace, tedy 2G.



Obr. 3.9 – Síť WWAN (Žákavcová, 2009)

3.5 BLUETOOTH TECHNOLOGIE

V této podkapitole je popsána technologie bluetooth, její standardy a protokoly, které poskytuje.



Bluetooth®

Bluetooth technologie pochází z roku 1994. Byla vytvořena švédskou firmou Ericsson, jako bezdrátová náhrada za sériové drátové rozhraní RS-232. Název "Bluetooth" pochází z 10. Století, byl odvozen od dánského krále Harald Blatand. Blatand se překládá do angličtiny jako bluetooth, tedy („modrozub“).

Řadí se do skupiny WPAN (*wireless personal area network*). Umožňuje spojení *point-to-point* (propojení dvou zařízení) nebo *point-to-multipoint* (propojení více zařízení).

Obr. 3.10 – Logo bluetooth (Černý, 2015a)

Tato technologie má mnoho výhod, ale i nevýhod. Mezi nevýhody patří například rušení, dosah a omezené frekvenční pásmo. Pracuje v pásmu ISM (tzv. nelicencované pásmo) s frekvencí 2,4 GHz, stejně jako Wi-Fi. Na druhou stranu nejsou zapotřebí kabely. Bluetooth je bezdrátová technologie krátkého dosahu. Vzdálenost přenosu závisí na verzi bluetooth (Černý, 2014).

Bluetooth technologie umožňuje výměnu dat pomocí rádiového přenosu. V této práci byl na straně robota použit bluetooth modul HC-05 a na druhé straně byla použita bluetooth technologie, která je zabudována přímo v notebooku.

3.5.1 Standardy

Bluetooth technologie je definována standardem IEEE 802.15.1. Tento standard umožňuje bezdrátovou komunikaci bez velkých nároků na spotřebu energie a konfiguraci.

BLUETOOTH V1.0 A V1.B

Jedná se o první verzi, která měla mnoho problémů s kompatibilitou. Postrádala schopnost jednoznačného přiřazení režimu master nebo slave.

BLUETOOTH V1.1

Tato verze byla schválena jako standard IEEE 802.15.1 v roce 2002. Odstranili se některé chyby, které předchozí verze vykazovali, a také byla přidána podpora pro nešifrované kanály.

BLUETOOTH V1.2

Umožňuje zpětnou kompatibilitu s předchozí verzí 1.1. Má rychlejší připojení a zlepšuje odolnost vůči rádiovému rušení.

BLUETOOTH V2.1 +EDR

Hlavní výhodou této verze je jednoduché párování a zvýšená bezpečnost.

BLUETOOTH V3.0 +HS

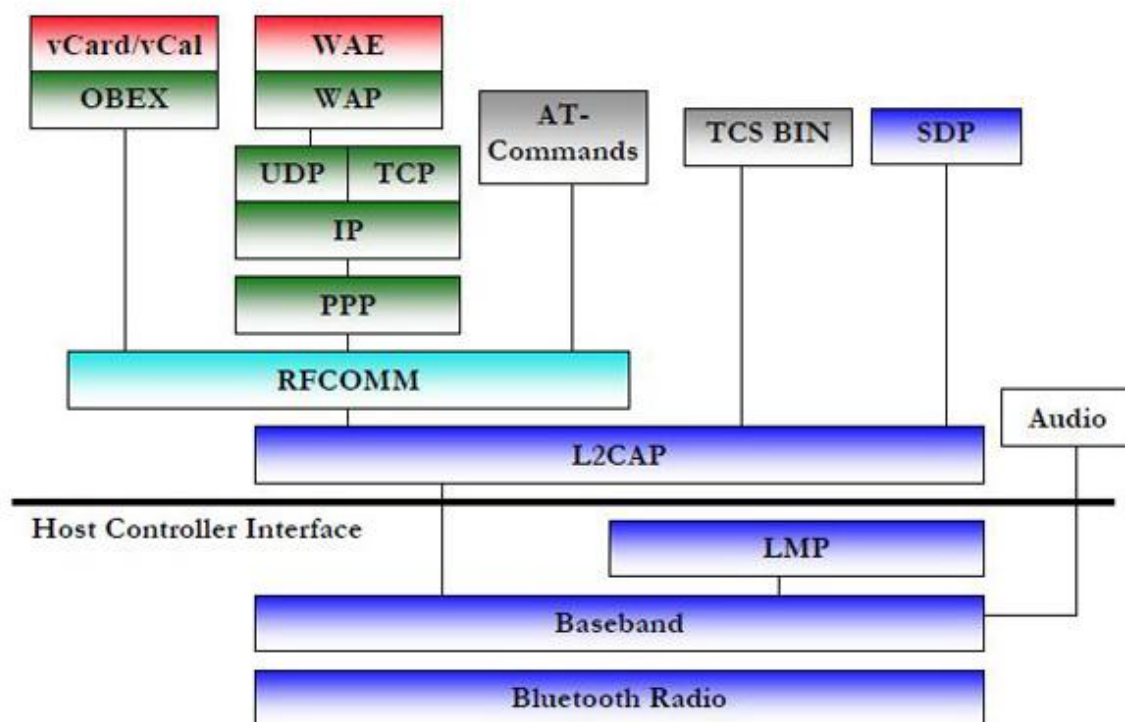
Nabízí vysokorychlostní přenos. Rychlost přenosu je srovnatelná s Wi-fi.

BLUETOOTH V4.0

Vydána v roce 2010. Snížení nároků na spotřebu energie.

3.5.2 Bluetooth architektura

Komunikační protokol bluetooth technologie v sobě zahrnuje všechny úrovně vrstev. Využívá i protokoly vyskytující se ve vyšších aplikačních vrstvách obr 3.11. Pro bluetooth jsou povinné tyto 3 protokoly LMP, L2CAP a SDP.



Obr. 3.11 – Bluetooth protokoly a vrstvy (Žákavcová, 2009)

LMP

LMP protokol je realizován na radiči a slouží k rádiovému spojení mezi dvěma zařízeními. Kontroluje a ukončuje spojení. Má na starost kontrolu párovacích klíčů, samotné párování a šifrování. Párovací klíč slouží ke spárování například počítače s telefon pomocí bluetooth. Musí se zadat stejný párovací klíč, jak v počítači, tak v telefonu.

L2CAP

Spojení dvou zařízení pomocí protokolů vyšší úrovně. Umožňuje také rozdělování a spojování paketů. Zajišťuje multiplexování protokolů (SDP, TCS, RFCOMM).

SDP

Tento protokol slouží k zjišťování služeb. Poskytuje aplikacím prostředky pro zjištění těchto služeb.

BLUETOOTH RADIO

Tato vrstva specifikuje frekvenční pásmo a také řídí vysílací výkon.

BASEBAND

Zajišťuje spojení mezi dalšími moduly bluetooth. Řídí komunikaci, synchronizaci a časové dělení kanálů.

RFCOMM

Radiofrekvenční komunikace, slouží jako náhrada kabelového protokolu. Poskytuje binární přenos dat. Slouží k zadávání AT příkazů, které jsou blíže popsány v praktické části této práce.

HCI

Poskytuje příkazové rozhraní pro řídicí pásma a také přístup ke konfiguračním parametrům.

PPP

Je to point-to-point protokol, který je orientovaný paketově. Používá se pro přenos paketů.

TCP

Bitově orientovaná funkce. Definiuje řízení a přenos dat.

WAP

Umožňuje zpřístupnění internetového obsahu mobilním zařízením.

OBEX

Potřebný k výměně dat. Využívá architekturu klient-server.

AT Commands

Příkazy, pomocí kterých lze změnit název modulu, režim atd.

Třídy bluetooth technologie

Bluetooth zařízení se dělí do tříd (class) podle výkonu a dosahu tab 3.1.

Tab. 3.1 - Třídy bluetooth technologie (Žákavcová, 2009)

Class	Maximální povolený výkon		Dosah
	mW	dBm	
Class 1	100	20	~ 100 m
Class 2	2,5	40	~10 m
Class 3	1	0	~ 1 m

3.5.3 Bluetooth modul HC-05

Modul HC-05 nabízí dva typy přenosů, synchronní a asynchronní. Tento modul také umožňuje nastavit dva módy master a slave. Master je řídicí stanice, která může obsloužit maximálně sedm podřízených (slave) stanic. Přenosová rychlost tohoto modulu je standardně

nastavena na 9 600 baudů. Nicméně tento modul podporuje více přenosových rychlostí, jak je patrné z tab. 3.2.

Výběr mezi módem master nebo slave se provádí pomocí AT příkazů tab. 3.3, stejně tak i změna přenosových rychlostí. Pomocí AT příkazů lze změnit i název bluetooth modulů, čehož bylo využito v následující kapitole.

Propojením dvou a více zařízení, které sdílejí stejný fyzický kanál, vznikne síť, která se nazývá *piconet*. Propojení více piconetů se nazývá *scatternet*, který umožňuje komunikaci mezi více než osmi zařízeními. Schéma propojení můžeme vidět na obr. 3.12 (Černý, 2015a).



Obr. 3.12 – Spojení piconet a scatternet (Žákavcová, 2009)

Tab. 3.2 - Přenosové rychlosti

Bluetooth modul HC-05
Přenosová rychlost, Bd
4 800
9 600
19 200
38 400
57 600
115 200
230 400
460 800
921 600
1 382 400

Tab. 3.3 - Tabulka AT příkazů

Příkaz	Popis
AT	Ověření funkčnosti spojení.
AT+RESET	Restartuje modul.
AT+VERSION	Verze modulu.
AT+HELP	Seznam všech příkazů.
AT+NAME	Vrátí/nastaví název modulu.
AT+PIN	Vrátí/nastaví párovací kód modulu.
AT+UART	Vrátí/nastaví přenosovou rychlost.
AT+CLEAR	Odstraní adresu.
AT+LADDR	Localní adresa.
AT+ROLE	Vrátí/nastaví režim (master, slave).
AT+DEFAULT	Obnoví nastavení.
AT+CMODE	Vrátí/nastaví možnosti připojení.
AT+CLASS	Vrátí/nastaví třídu modulu.

4 PRAKTICKÁ REALIZACE

V této kapitole jsou popsány softwary MATLAB a WIRING, pomocí kterých se vytvořil komunikační protokol a naprogramovala bezdrátová komunikace. Dále je zde popsána platforma Arduino, která řídí celý robot a nezbytné kroky, které jsou zapotřebí pro navázání komunikace.

4.1 MATLAB MATHWORKS

MATLAB představuje nadřazený systém, který přijímá data z kamerového systému. Následně se data zpracují, tj. je určena struktura bludiště a poloha robotů, poté je provedeno plánování a převod plánů na sekvence instrukcí. Na základě vygenerovaných plánů jsou pak vytvořeny instrukce pro roboty, které jsou posílány robotům pomocí navrženého komunikačního protokolu.



Obr. 4.1 – Logo MATLAB

MATLAB byl vyvinut v roce 1984 společností The MathWorks, Inc. v USA.

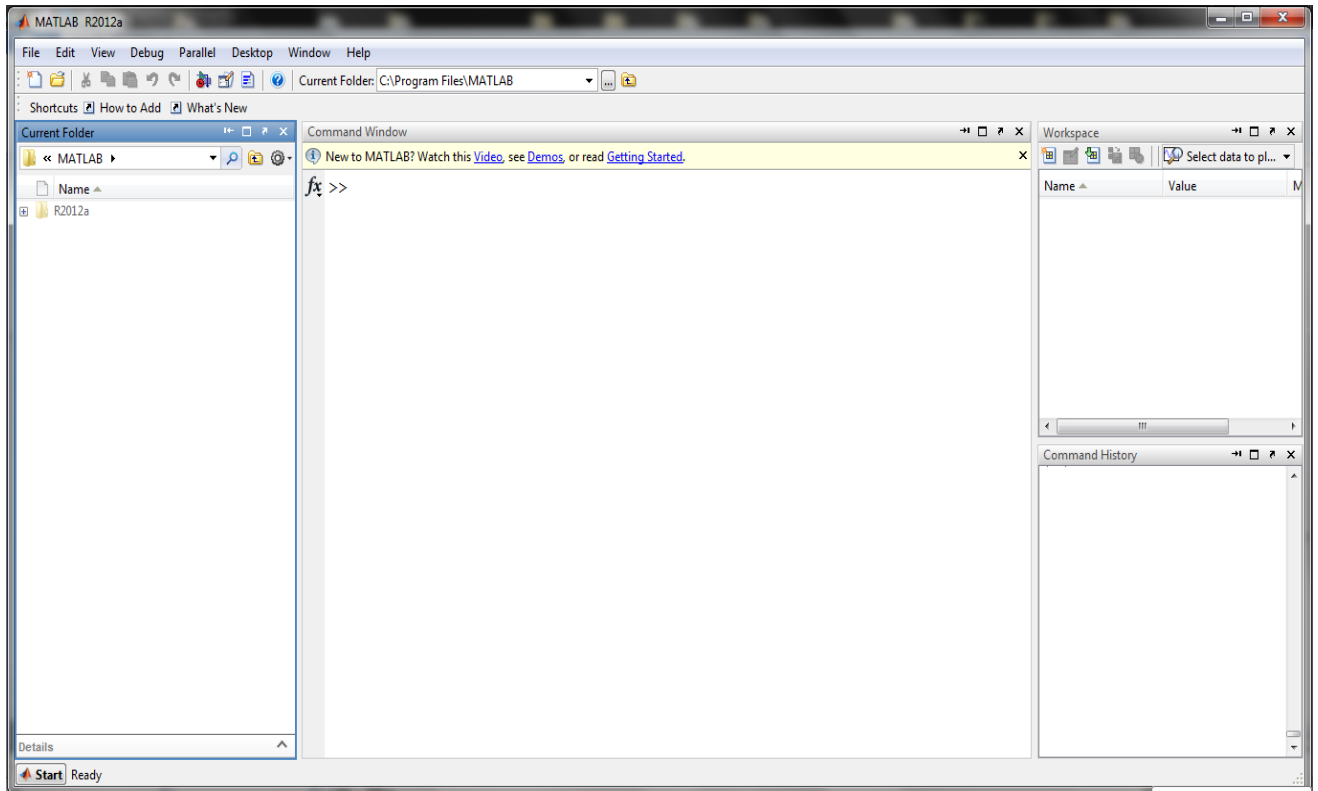
Název MATLAB byl odvozen z anglického sousloví „Matrix Laboratory“ (volně přeloženo „maticová laboratoř“).

Již z názvu vyplývá, že tento software umožňuje provádět operace s maticemi. Obsahuje spoustu vestavěných funkcí, stačí jen zadat název funkce a příslušné parametry. Funkce MATLABu jsou velice podobné programovacímu jazyku C++.

MATLAB také umožňuje tvorbu programů, vytváření aplikací s grafickým rozhraním, toto prostředí se nazývá GUIDE a také tvorbu simulací v nadstavbě SIMULINK, čímž se nebudeme zabývat.

Použitá verze MATLABu pro tuto práci je R2012a, její grafické prostředí je zobrazeno na obr. 4.2. Jak je z obrázku patrné prostředí programu MATLAB se skládá ze čtyř základních bloků. Jsou to: adresář současného umístění, který lze libovolně měnit (Current Folder), příkazové okno (Command Window), historie příkazů (Command History) a blok, kde můžeme vidět, které proměnné uchováváme v paměti a jejich bližší informace (Workspace), jako je například datový typ a hodnota (Dušek, 2000).

Pro účely bakalářské práce byl využit „Editor“, ve kterém lze vytvářet programy (skripty) i funkce, ty můžeme posléze libovolně ukládat a spouštět. Také byl využit Instrument Control Toolbox, který obsahuje funkce pro práci se sériovým portem.



Obr. 4.2 – Grafické prostředí MATLAB R2012a

4.2 ARDUINO

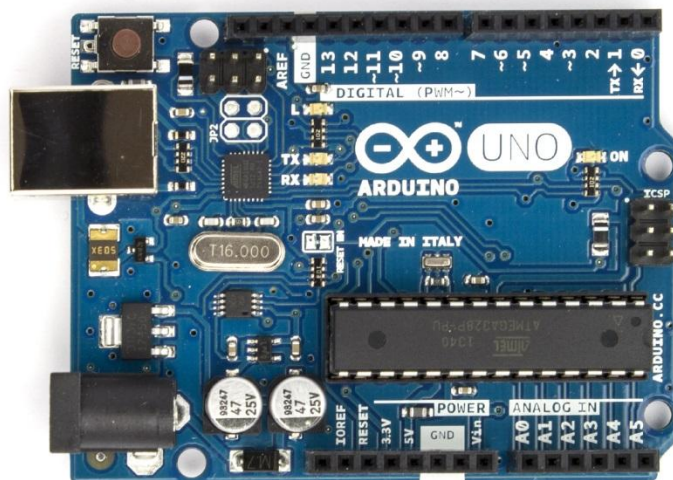
Arduino je otevřená elektronická platforma, založená na jednoduchém hardware a software. Desky Arduino obsahují 8-bitové mikroprocesory AVR od firmy Atmel. Arduino může být použito k vytváření samostatných interaktivních zapojení nebo může být připojeno k softwaru na počítači. Obsahuje IDE (integrované vývojové prostředí), které je založeno na jazyku WIRING.

WIRING je programovací jazyk podobný jazyku C++, který má vylepšené knihovny. Vychází z open source projektu, který se nazývá Processing (flexibilní software). Pro účely této práce byla využita platforma Arduino Uno verze R2 (Malý, 2010).

4.2.1 Arduino Uno

Arduino Uno obr. 4.3, obsahuje jednočipový mikrokontrolér ATmega328, který je typu RISC (Reduced Instruction Set Computer) a hardvardské architektury. ATmega328 obsahuje 32 kB FLASH paměť pro uložení programu (0,5 kB používá zavaděč), 2 kB vnitřní SRAM a 1 kB EEPROM. Dále obsahuje periferní funkce, jako jsou časovače / čítače. Tento mikrokontrolér má propustnost 20 MIPS (milión instrukcí za sekundu) při taktu 20 MHz.

Samotná deska má 14 digitálních vstupních / výstupních pinů, 6 analogových vstupů, které jsou označeny A0 – A5. Připojení k počítači probíhá přes USB, tím je zajištěno i napájení celé platformy. Alternativním zdrojem je 9 V baterie. To se používá zejména v případech, kdy je komunikace zajištěna bezdrátově. Napájení 5 V nebo 3,3 V.



Obr. 4.3 – Platforma Arduino Uno

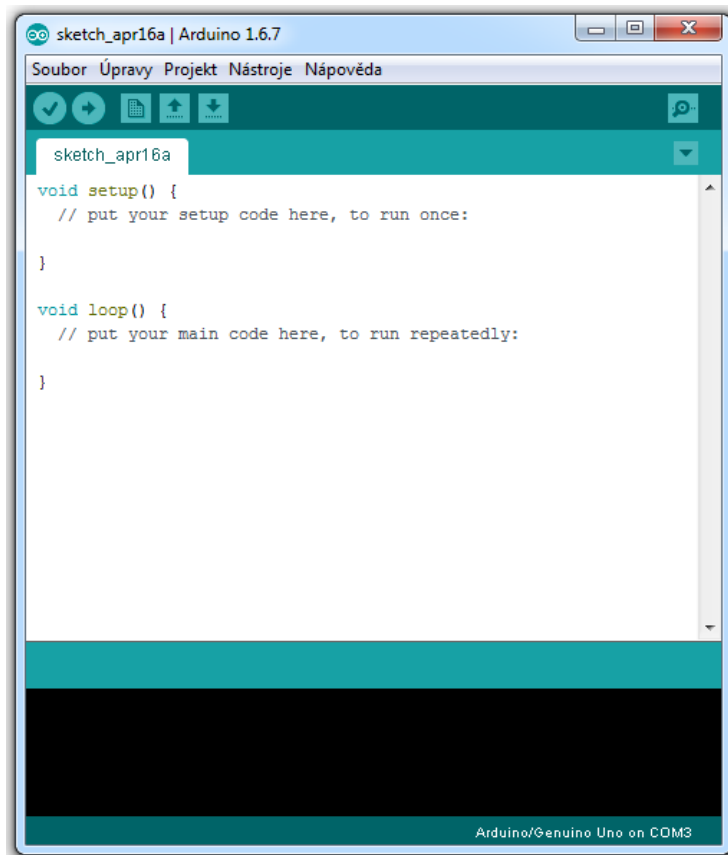
4.2.2 Vývojové prostředí Arduina

Integrované vývojové prostředí je tzv. open source (volně ke stažení na internetu). Jak bylo již výše zmíněno, k programování slouží jazyk WIRING.

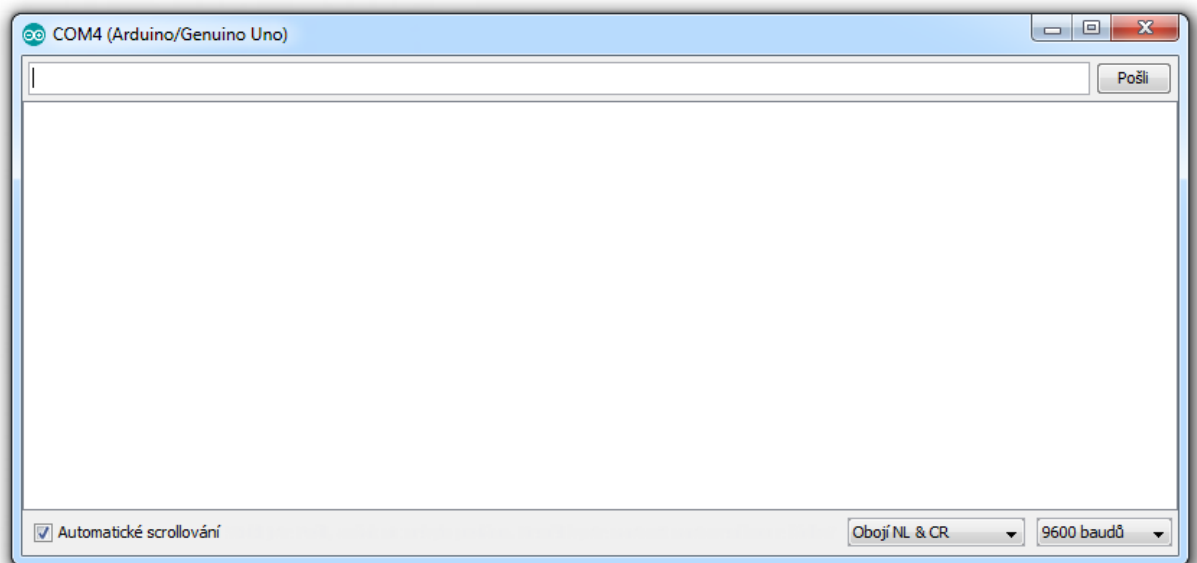
Prostředí Arduino se skládá ze tří základních částí, jak je zobrazeno na obr. 4.4 níže. Horní „lišta“ obsahuje menu a základní funkce, těmi jsou: kompilace programu, nahrání programu do Arduina, nový program, načtení dříve vytvořených programů, uložení stávajícího programu a sériový monitor. V sériovém monitoru obr. 4.5 se zobrazují data, které jsou posílány nebo přijímány pomocí sériového portu. Sériový monitor lze otevřít pouze po připojení platformy Arduino k počítači.

V prostřední části se vytváří program (sketch). Můžeme si všimnout, že se dělí na dva základní bloky. Prvním z nich je *void setup*. Mezi jehož složené závorky se píše kód, který se provede pouze jednou na začátku programu. To znamená buď po připojení napájení, zmáčknutí tlačítka restart nebo nahrání kódu do Arduina. V případě sériové komunikace se zde uvádí rychlost přenosu v baudech. Druhým blokem je *void loop*, do jehož složených závorek se zapisuje kód, který se bude opakovat neustále dokola až do odpojení napájení. Tyto základní bloky musí být v programu vždy, jinak by došlo k chybovému hlášení.

Poslední část slouží k informačním a chybovým výpisům z běhu programu. Můžeme zde vidět jaký typ Arduina používáme a k jakému portu počítače je připojen. V tomto případě se jedná o Arduino Uno a port COM 03 (Malý, 2011).



Obr. 4.4 – Vývojové prostředí Arduino



Obr. 4.5 – Sériový monitor Arduino

4.3 PRAKTICKÉ ŘEŠENÍ

V této podkapitole je popsáno zprovoznění bezdrátové komunikace mezi prostředím MATLAB a Arduino Uno s použitím bluetooth modulů HC-05.

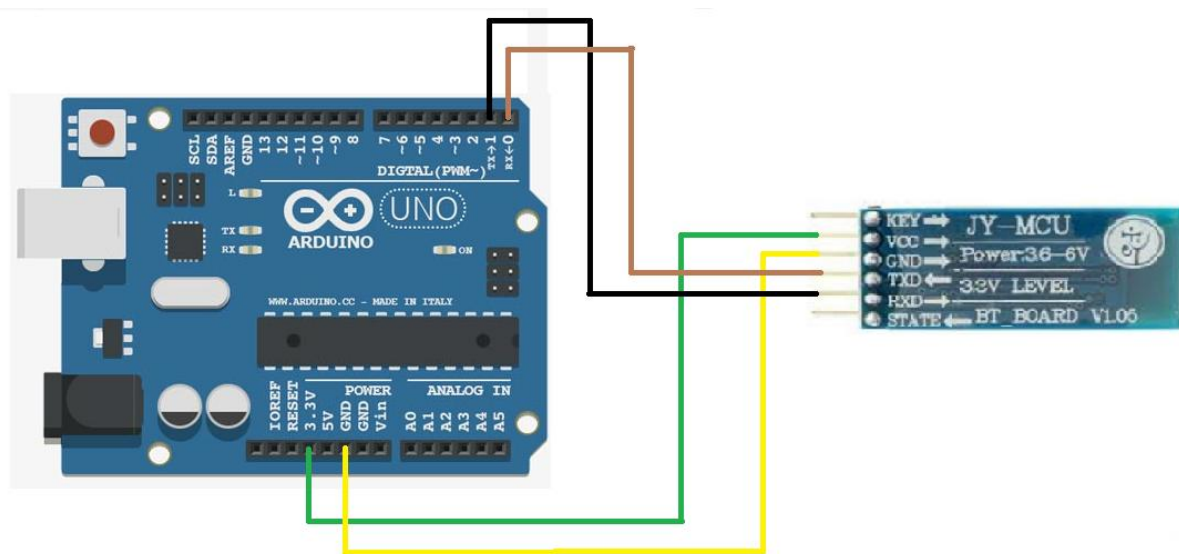
4.3.1 Připojení modulů HC-05 k Arduino Uno platformě

Nejdříve než začneme psát samotný program, musíme připojit bluetooth modul HC-05 k Arduino platformě. Můžeme se setkat se zapojením přes odporový dělič. Nicméně v mém případě to nebylo potřeba, stačilo připojit napájení bluetooth na 3,3 V na Arduino, protože moduly HC-05 umožňují napájení 3,3 V i 5 V, stejně tak i Arduino Uno platforma. Poté se musí propojit pin Tx (vysílač) na Arduino s Rx (přijímač) na bluetooth a opačně. Následně propojit zem (GND), (Černý, 2015b).

Schéma zapojení je na obr. 4.7.



Obr. 4.6 – Bluetooth modul HC-05



Obr. 4.7 – Připojení bluetooth modulu HC-05 k Arduino Uno platformě

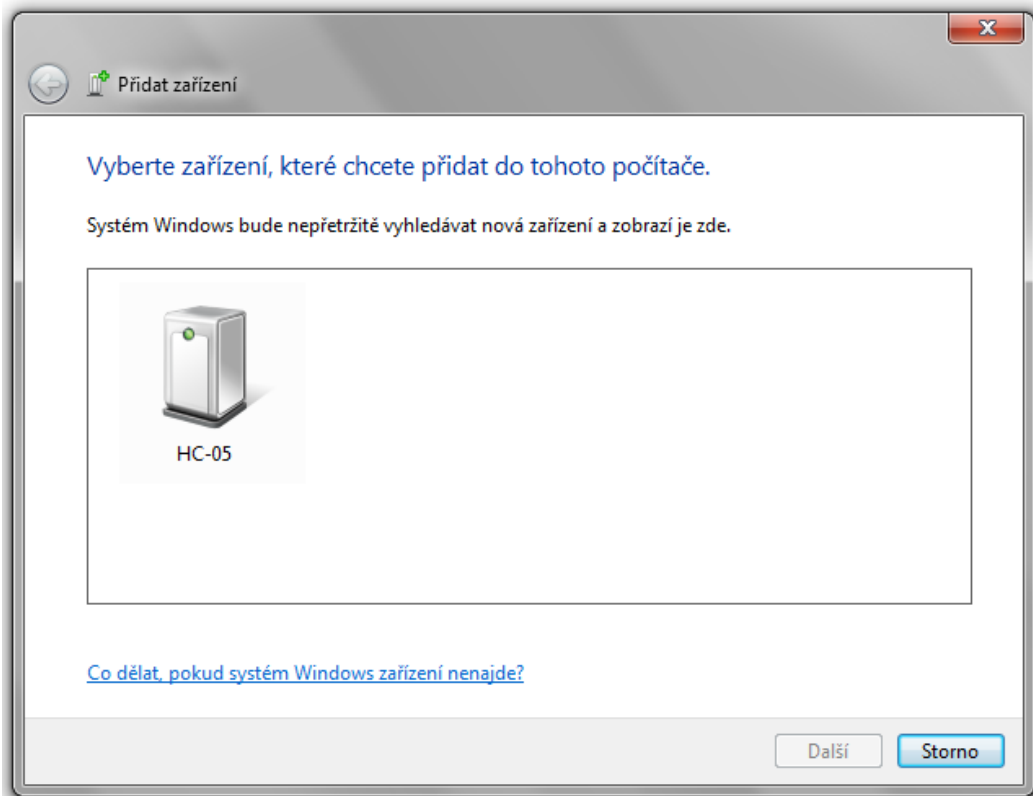
4.3.2 Spárování zařízení

Dalším krokem je spárování zařízení, tedy bluetooth modulů HC-05 s bluetooth technologií, která je přímo zabudována v notebooku, který slouží k posílání poslušností do Arduina.

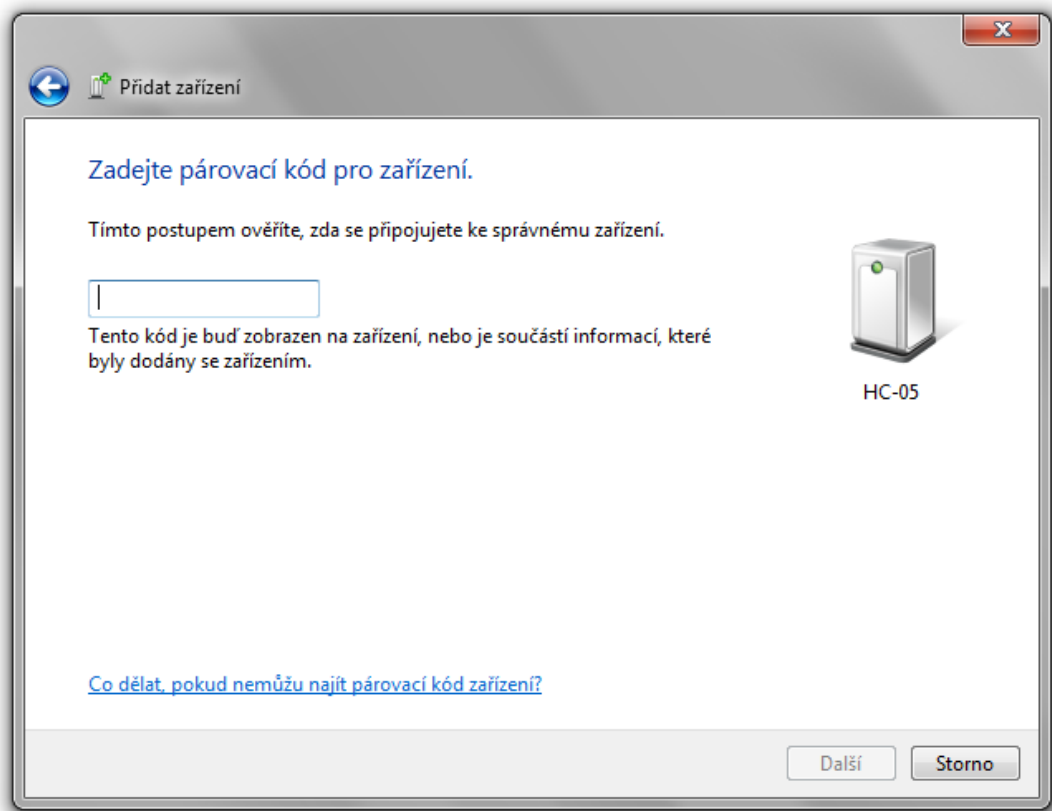
Spárovat zařízení není vůbec složitý proces. Stačí na našem počítači kliknout na ikonu bluetooth pravým tlačítkem, která se běžně nachází v hlavní nabídce v pravém rohu obr. 4.8. Poté se klikne na přidat zařízení. V tuto chvíli nám vyskočí dialogové okno, které nám ukáže, které zařízení jsou k dispozici obr. 4.9 níže. Po zvolení daného zařízení, se kterým chceme počítač spárovat nám vyskočí další dialogové okno obr. 4.10, kde stačí zadat párovací kód. U bluetooth HC-05 to bývá standardně 1234. Poté co zadáme párovací kód, můžeme začít pracovat s bluetooth.



Obr. 4.8 – Hlavní nabídka



Obr. 4.9 – Přidat zařízení bluetooth



Obr. 4.10 – Zadání párovacího kódu bluetooth

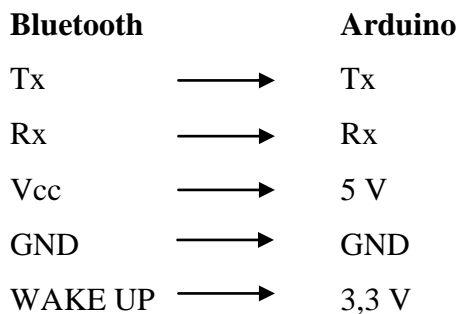
4.3.3 Nastavení v Arduinu

Jestliže jsou bluetooth technologie spárovány může se připojit platforma Arduino Uno k počítači, poté se musí nastavit parametry v sériovém monitoru. O sériovém monitoru jsem se zmínil již dříve, nachází se v horní liště pod ikonou lupy. Zde se musí nastavit rychlost přenosu a řádkování. V této práci, byly využity přenosové rychlosti z tab. 3.2 a řádkování bylo nastaveno na „Obojí NL & CR“.

Důležité je mít nastavené stejné přenosové rychlosti, jak v sériovém monitoru, tak přímo v programu, jinak dochází ke špatnému přenosu a výpisu dat. To znamená, když bude zvolena přenosová rychlost 115 200 baudů, tak to musí být zvolené v sériovém monitoru a v programu musí být funkce *Serial.begin(115200)*.

Nicméně, tímto způsobem by byla umožněna komunikace jen s jedním robotem, protože moduly HC-05 mají standardně nastaven název „HC-05“. Proto u zbývajících robotů se musí změnit název. To se provede následovně:

1. Nejdříve se musí změnit zapojení obr. 4.11.



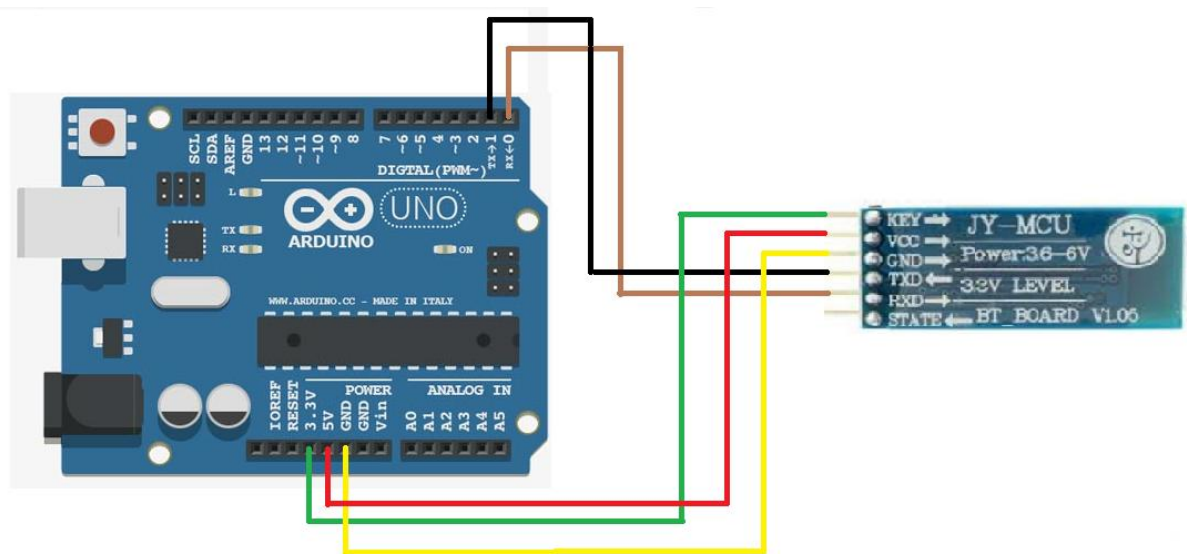
2. Nyní se připojí Arduino k počítači a nahraje prázdný program.
3. V sériovém monitoru se nastaví přenosová rychlost 38 400 baudů, protože tato přenosová rychlost souží k zadávání AT příkazů.
4. Dioda na bluetooth HC-05 začne pomalu blikat (jednou za 2 sekundy), to indukuje stav, že modul HC-05 je v příkazovém režimu a můžeme začít posílat AT příkazy.
5. Do sériového monitoru nyní napíšeme příkaz „AT+NAME?“ bez uvozovek. Vzápětí nám to vrátí název modulu v tomto případě: +NAME:HC-05.
6. Změna názvu se provede příkazem „AT+NAME=HC2“. Arduino odpoví OK.
HC2 je název druhého modulu.

Tímto způsobem se pokračuje až do té doby, než se všechny názvy bluetooth modulů budou lišit. Avšak změna názvu neproběhne ihned, ale až po vypnutí napájení a vybití kondenzátorů. Nyní si můžeme zase ověřit název modulu pomocí předchozích příkazů.

Dále je zapotřebí změnit přenosovou rychlost modulů HC-05, která je standardně 9 600 baudů. To se provede stejným způsobem, jen se použije příkaz „AT+UART=230400,0,1“. Kde první parametr je požadovaná přenosová rychlost. Druhý parametr je stop bit a třetí je parita.

U přenosové rychlosti 230 400 baudů musí být nastavena parita na 1, protože jinak dochází k chybnému výpisu dat. Parita slouží k detekci chyby ve slově. U nižších rychlostí není potřeba mít nastavenou paritu, proto nastavení vypadá například takto „AT+UART=115200,0,0“.

Jestliže má každý bluetooth modul jiný název a přenosová rychlost je nastavena na 230 400 baudů, změní se znovu schéma zapojení podle obr. 4.7, které slouží k odesílání a přijímání dat. A poté se nahraje příslušný program.



Obr. 4.11 – Schéma zapojení pro uvedení modulu HC-05 do příkazového režimu

4.3.4 Nastavení v MATLABu

Jelikož bylo nastavené a zapojené vše nezbytné pro komunikaci, můžeme přejít k samotnému programu.

Prvním krokem v MATLABu je vytvoření bluetooth objektu. Bluetooth objekt se vytváří příkazem *Bluetooth('HC-05', 1)*. Kde první parametr HC-05 je název bluetooth, které se používá a druhý parametr je tzv. channel neboli kanál, který bývá u bluetooth modulů HC-05 standardně nastaven na 1.

Druhým krokem je „otevření“ komunikace, čehož docílíme příkazem *fopen(b)*, kde *b* je mnou zvolený název sériového portu. Navázání komunikace s druhým robotem se provede stejným způsobem, jen změníme název bluetooth, takže to bude vypadat takto *Bluetooth('HC2', 1)*.

Jestliže jsme navázali úspěšně komunikaci, můžeme přejít k odesílání. Data se odesílají příkazem *fprintf(b, a)*, kde první parametr je bluetooth objekt a druhý parametr jsou data, která se odesílají. V tomto případě tedy posloupnost příkazů s přiděleným identifikátorem.

Při testech se data posílali střídavě, to znamená, že se nejdříve poslala jedna posloupnost prvnímu robotovi a jedna druhému robotovi atd. Test se snaží napodobit skutečnost, kdy by mohlo dojít k přepsání posloupnosti příkazů, kterou mají uloženou v paměti a tím i ke kolizi. Tímto způsobem se jim poskytne nějaký čas na dokončení posloupnosti, kterou mají již uloženou v paměti.

5 EXPERIMENTÁLNÍ OVĚŘENÍ

Byl proveden experiment, kdy se ověřovala přenosová rychlost a spolehlivost zvolené technologie, tedy bluetooth a její dosah. Posílala se sekvence dat opakovaně z MATLABu, při různých vzdálenostech a různých přenosových rychlostech, která se následně vypsala v sériovém monitoru Arduina. Tento proces se opakoval 100 krát, tím vznikla tabulka o 100 řádcích. Následně byly data z MATLABu a Arduina porovnány, jestli vzájemně souhlasí, z toho byla odvozena spolehlivost vybraných přenosových rychlostí.

V experimentu byl použit pouze jeden bluetooth modul a jedna Arduino Uno platforma, protože všechny bluetooth moduly HC-05 fungují stejně, tudíž by se museli zbytečně porovnávat data navíc.

Jak je z experimentu patrné, byly vyzkoušeny tyto přenosové rychlosti (4 800, 9 600, 38 400, 57 600, 115 200 a 230 400 baudů). Spolehlivost se nějak zásadně nelišila, proto nejvhodnější přenosovou rychlostí je 230 400 baudů. Tato přenosová rychlost je i zároveň maximální přenosová rychlost, při které bylo možné posílat data bez problémů. Tato přenosová rychlost byla zvolena i z toho důvodu, že sériový monitor Arduina nepodporuje vyšší přenosovou rychlost.

Nicméně u bluetooth modulů HC-05 se dají měnit i třídy tzv. class tab. 3.1 a tím se mění i jejich dosah. Modul HC-05 je standardně nastaven na class 2, proto se maximální dosah v experimentu pohyboval okolo 10 m.

Tyto moduly umožňují i nastavení class 1, u této třídy je předpokládaný dosah až 100 m, ale za cenu větší spotřeby energie. Vzhledem k tomu, že dosah 10 m je pro tuto laboratorní úlohu dostačující, tak moduly byly nastaveny na class 2.

Výsledky experimentu jsou v tab. 5.1.

Tab. 5.1 - Tabulka spolehlivosti

Vzdálenost, m	Přenosová rychlost, Bd					
	4 800	9 600	38 400	57 600	115 200	230 400
	Spolehlivost, %					
1	100	100	100	100	100	100
2	100	100	100	100	100	100
3	100	100	100	100	100	100
4	100	100	100	100	100	100
5	100	100	100	100	100	100
6	100	100	100	100	100	100
7	100	100	100	100	100	100
8	100	100	100	100	100	100
9	100	100	100	100	100	100
10	100	100	100	100	100	100
11	47	47	48	45	46	46

6 ZÁVĚR

Cílem této bakalářské práce byl návrh a realizace komunikačního protokolu pro obousměrný bezdrátový přenos dat mezi MATLABem a robotickými vozítky a následná verifikace. V rámci vývoje robota nebylo nutné řešit obousměrný přenos, ačkoliv to MATLAB i Arduino Uno umožňuje, protože data se dají vypsat pro kontrolu v obou softwarech.

Komunikační protokol byl vytvořen v softwarech MATLAB a WIRING. MATLAB zároveň představuje nadřazený systém, který zajišťuje trajektorii pohybu robotů na základě dat přijatých z kamerového systému a požadavků uživatele. Tím vznikne naplánovaná cesta, kterou má robot projet z počátečního bodu do cílového bodu. Tyto informace se dále rozdělily na posloupnost pěti příkazů, kde každé číslo reprezentuje určitou akci, kterou mají roboty vykonat. Následně se k této posloupnosti přidělil identifikátor, tím vznikla posloupnost šesti příkazů. Jako identifikátor byla zvolena celá záporná čísla od -1 do -5. Tímto způsobem se předešlo problému, kdy by se mohli přijaté data v robotech opakovat.

Bezdrátový přenos byl realizován pomocí bluetooth technologie, konkrétně pomocí modulů HC-05. Zapotřebí byly také platformy Arduino Uno a software WIRING.

V sériovém monitoru softwaru WIRING, byla provedena změna názvů bluetooth modulů HC-05 pomocí AT příkazů. Změna názvů byla provedena z toho důvodu, aby byla možnost současně nasadit až pět robotů. Názvy bluetooth modulů byly zvoleny tak, aby se lišily, jinak by docházelo k problémům s navázáním komunikace, protože MATLAB by nevěděl, který bluetooth modul má kontaktovat.

Pro přenos dat byla zvolena přenosová rychlost 230 400 baudů a nastavení bluetooth modulů na class 2. Tato přenosová rychlost byla zvolena z toho důvodu, že se jedná o maximální rychlost, kterou sériový monitor Arduina podporuje. Nicméně při této rychlosti musí být nastavena parita, jinak dojde k chybnému přenosu dat.

Testován byl také dosah zvolené technologie při nastavení modulů na class 2 a také při různých přenosových rychlostech. Dosah byl 10 m při všech přenosových rychlostech, což je pro tuto laboratorní úlohu dostačující a spolehlivost byla do 10 m vždy 100%.

LITERATURA

- ČERNÝ, M. 2015a. Programování Bluetooth modulů HC-05 (1). *RoboDoupě* [online]. [cit. 2016-04-22]. Dostupné z: <http://robodoupe.cz/2015/programovani-bluetooth-modulu-hc-05-a-hc-06/>
- ČERNÝ, M. 2015b. Programování Bluetooth modulů HC-05 (2). *RoboDoupě* [online]. [cit. 2016-04-22]. Dostupné z: <http://robodoupe.cz/2015/programovani-bluetooth-modulu-hc-05-2/>
- ČERNÝ, M. 2014. Ovládání pomocí Bluetooth – pro Arduino. *RoboDoupě* [online]. [cit. 2016-04-22]. Dostupné z: <http://robodoupe.cz/2014/ovladani-pomoci-bluetooth-pro-arduino/>
- HC-05 Bluetooth module*. 2010. Datasheet. [online]. [cit. 2016-04-22]. iteaddstudio. Dostupné z: http://www.robotshop.com/media/files/pdf/rb-ite-12-bluetooth_hc05.pdf
- DUŠEK, F. 2000. *MATLAB a SIMULINK – úvod do užívání*. Univerzita Pardubice [cit. 2016-04-22]. 147 s. ISBN 80-7194-273-1. Dostupné z: http://bellman.zcu.cz/~mcech/simulink_navod.pdf
- MALÝ, M. 2011. Ovládání Arduina v reálném čase z počítače. *Root* [online]. [cit. 2016-04-23]. ISSN 1212-8309. Dostupné z: <http://www.root.cz/clanky/ovladani-arduina-v-realnem-case-z-pocitace/>
- MALÝ, M. 2010. Arduino: jak pro něj začít programovat. *Root* [online]. [cit. 2016-04-23]. ISSN 1212-8309. Dostupné z: <http://www.root.cz/clanky/ovladani-arduina-v-realnem-case-z-pocitace/>
- SIEGWART, R; NOURBAKSH, I; SCARAMUZZA, D. 2011. *Introduction to Autonomous Mobile Robots. Second edition*. London: The MIT Press. 472 s. ISBN 978-0262015356
- ŠÍMA, M. 2000. *Plánování cesty robota ve spojitém prostředí*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství. 63 s. Dostupné z: http://autnt.fme.vutbr.cz/szz/2006/DP_Sima.pdf
- ŠKRABÁNEK, P.; MARIŠKA M.; DOLEŽEL, P. 2015a. The time optimal path-planning of mobile robots motion respecting the time cost of rotation. *Strbske Pleso: New York: IEEE*. [cit. 2016-04-23]. ISBN 978-1-4673-6627-4.
- ŠKRABÁNEK, P. 2015b. *Labyrinth arrangement analysis based on image processing. Mendel: Dordrecht:Springer*. [cit. 2016-04-23]. ISBN 978-3-319-19823-1.
- VLČEK, J. *Přenos dat* [online]. [cit. 2016-04-22]. Dostupné z: file:///C:/Users/Vojcek/Downloads/000102_mereni_a_sber_dat_pomoci_pc.pdf
- VODIČKA, P. 2016. *Vytvoření řídicího softwaru pro robotické vozítko*. Pardubice: Univerzita Pardubice. Fakulta elektrotechniky a informatiky. Vedoucí: Pavel Škrabánek
- ŽÁKAVCOVÁ, M. 2009. *Tvorba bezdrátové domácí sítě Wi-fi a možnost sdílení pomocí technologie bluetooth*. Bakalářská práce. Praha: Vysoká škola ekonomická v Praze. Fakulta informatiky a statistiky. 54 s. Dostupné z: <http://info.sks.cz/www/zavprace/soubory/68700.pdf>

PŘÍLOHY

A - CD

Příloha k bakalářské práci

Návrh a realizace komunikačního protokolu a softwaru pro bezdrátový přenos
dat mezi robotickým vozíkem a MATLABem

Vojtěch Ihnát

CD

Obsah

- 1 Text bakalářské práce ve formátu PDF
- 2 Řídící program vytvořený v prostředí MATLAB
- 3 Program vytvořený v prostředí WIRING