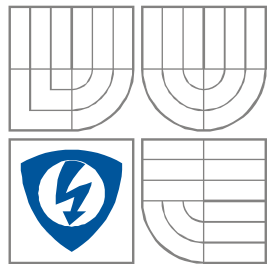


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SENZORICKÝ SYSTÉM PRO ROBOTICKÝ PODVOZEK

SENSOR SYSTEM FOR ROBOTIC CHASSIS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAKUB HŮLKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. STANISLAV KLUSÁČEK

BRNO 2012

SENZORICKÝ SYSTÉM PRO ROBOTICKÝ PODVOZEK

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou řízení robotu. Vyberte vhodný univerzální podvozek a zvolte senzorický systém pro pohyb robotu v průmyslovém prostředí. Pro zvolené řešení navrhnete řídicí systém, který zajistí obsluhu všech senzorů, ovládání motoru a dokáže komunikovat se vzdáleným operátorem.

V bakalářské práci se předpokládá realizace hardware navrženého robotického podvozku, implementace software pro zvolené řešení a vytvoření ovládacího programu pro operátorské řízení pomocí PC. Práce bude realizována ve spolupráci s UTEE.

DOPORUCENÁ LITERATURA:

- [1] Novák, P. Mobilní roboty - pohony, senzory, řízení. Praha : BEN – technická literatura, 2005. ISBN 80-7300-141-1
- [2] Mikroprocesorová technika [online]. 2009 [cit. 2010-05-20]. BMPT. Dostupné z WWW:<<http://www.urel.feec.vutbr.cz/BMPT/index.php?strana=5&lang=CS>>.

Abstrakt

Tato bakalářská práce se zabývá výběrem vhodného robotického podvozku pro průmyslové prostředí. Tento podvozek je dále rozšířen o senzorický systém, který zajišťuje interakci robotu s okolím. Pro navrženou koncepci je vytvořena vlastní řídicí deska, která obstarává obsluhu senzorů a ostatních periferií. Dále je zde popsán vývoj programu do PC pro operátorské řízení robotu.

Klíčová slova

Robot, senzor, řídicí, PIC, mikroC, autonomní

Abstract

This thesis deals with selection of a suitable robotic undercart for industrial environments. This undercart is further enhanced by a sensory system, that provides interaction with the robot's surroundings. For the proposed scheme is created its own control board, that performs service sensors and other peripherals. There is also described development of the program to your PC for operator control the robot.

Keywords

Robot, sensor, control, PIC, mikroC, autonomus

Bibliografická citace:

HŮLKA, J. SENZORICKÝ SYSTÉM PRO ROBOTICKÝ PODVOZEK. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 59s. Vedoucí bakalářské práce byl Ing. Stanislav Klusáček.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma SENZORICKÝ SYSTÉM PRO ROBOTICKÝ PODVOZEK jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **28. května 2012**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Stanislavovi Klusáčkovi a konzultantovi Ing. Martinu Friedlovi za jejich účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: 28.května 2012

.....
podpis autora

Obsah

1 Úvod.....	10
2 Robotické podvozky.....	11
2.1 Typy robotických podvozků.....	11
2.2 Přehled podvozků na trhu.....	15
2.3 Výběr podvozku.....	17
3 Senzory prostředí.....	18
3.1 Druhy senzorů.....	18
3.2 Přehled komerčních senzorů.....	22
3.3 Návrh senzorického systému.....	24
4 Řídící DESKA.....	26
4.1 Požadavky na řídicí jednotku.....	26
4.2 Zvolený procesor.....	28
4.3 Schéma řídicí desky.....	30
4.4 Seznam součástí.....	31
4.5 Deska plošných spojů.....	32
5 Programové řešení.....	34
5.1 Celková koncepce.....	34
5.2 Druhy zpráv.....	35
5.3 Firmware PIC16F877A.....	37
5.4 Program PC.....	51
6 Závěr.....	52
6.1 Návrhy na vylepšení.....	52
6.2 Foto hotového robota.....	53

Seznam ilustrací

Obr. 1: Diferenciální podvozek [5].....	11
Obr. 2: Synchronní podvozek [6].....	12
Obr. 3: Trojkolový podvozek.....	12
Obr. 4: Ackermanův podvozek [8].....	13
Obr. 5: Všesměrový podvozek.....	13
Obr. 6: Pásový podvozek [3].....	14
Obr. 7: Kráčeující podvozek[12].....	14
Obr. 8: UMU-01 [3].....	15
Obr. 9: TANK-01 [3].....	15
Obr. 10: TANK-02 [3].....	15
Obr. 11: TANK-03 [3].....	16
Obr. 12: MOB-02 [3].....	16
Obr. 13: MOB-03 [3].....	16
Obr. 14: Taktilní senzor [7]	18
Obr. 15: Infračervený senzor [7].....	19
Obr. 16: Odrazový triangulační infrasenzor [9].....	20
Obr. 17: Ultrazvukový senzor[10].....	21
Obr. 18: OPB704G [3].....	22
Obr. 19: OTR-1A [3].....	22
Obr. 20: GP2Y0A21 [3].....	22
Obr. 21: GP2Y0D805 [3]	23
Obr. 22: SRF02 [3].....	23
Obr. 23: SRF05 [3].....	23
Obr. 24: Blokové schéma navrženého senzorického systému.....	25
Obr. 25: Blokové schéma robotu.....	27
Obr. 26: Mikroprocesor PIC16F877A.....	28
Obr. 27: Schéma řídicí desky.....	29
Obr. 28: Motiv plošného spoje.....	31
Obr. 29: Osazovací výkres.....	32
Obr. 30: Zjednodušený pohled na komunikaci.....	33
Obr. 31: Způsob programování PIC16F877A.....	35
Obr. 32: Ukázka prostředí MikroC for PIC.....	36
Obr. 33: Ukázka programu ASIX UP.....	37
Obr. 34: Programátor PRESTO.....	38
Obr. 35: Řízení serva [13].....	39
Obr. 36: Blokové schéma Timer1.....	40
Obr. 37: SRF05 připojení [14].....	40
Obr. 38: SRF05 - časování signálů [14].....	41

Obr. 39: Závislost počtu průběhů smyčkou na vzdálenosti objektu.....	42
Obr. 40: GP2Y0A21 - závislost výstupního napětí na vzdálenosti.....	43
Obr. 41: Graf závislosti vzdálenosti na hodnotě z výstupu A/D převodníku.....	43
Obr. 42: Připojení H-můstků.....	44
Obr. 43: Diagram hlavní smyčky.....	45
Obr. 44: Diagram automatického režimu.....	46
Obr. 45: Diagram pohybu vzad.....	47
Obr. 46: Ovládací program v PC.....	48
Obr. 47: Hotový robot.....	50

Seznam tabulek

Tabulka 1: Základní parametry PIC16F877A [11].....	28
Tabulka 2: Konfigurace RS232.....	33
Tabulka 3: Přehled povelů pro robot.....	34
Tabulka 4: Zpráva od robota.....	34

1 ÚVOD

V dnešní době lidé vykonávají spoustu životu nebezpečných prací. Jednou z nich je například zneškodňování nevybuchlé munice pyrotechniky. Jejich úkoly a jiné rizikové činnosti dnes pomalu začínají přebírat roboti. Ať už se jedná o plně autonomní stroje nebo dálkově řízené operátorem. V této práci se pokusíme navrhnout koncepci robotického podvozku, který by v budoucnu mohl být na tyto nebezpečné činnosti nasazován.

Jelikož se jedná o rozsáhlou problematiku, na návrhu a realizaci této práce spolupracuji společně s kolegou Jozefem Humajem. Ten se zaměřil na zajištění bezdrátové komunikace a přenos videa z robota, dále obstaral návrh napájecí části společně s H-můstkou. Můj úkol spočíval ve výběru podvozku, senzorického systému a vývojem řídicí elektroniky pro vybrané řešení. Jednotlivé úkoly jsme si rozdělili podle zkušeností s danými tématy. Celková koncepce a rozdělení je ukázáno ve čtvrté kapitole, kde se zabýváme požadavky na řídicí jednotku.

V práci se budeme zabývat různými typy robotických podvozků a jejich vlastnostmi, přičemž se zaměříme na volbu vhodného podvozku do předpokládaného průmyslového prostředí. Při výběru je důležité si uvědomit, jak takové prostředí bude vypadat a určit jaké vlastnosti od něho očekáváme. Je veliký rozdíl mezi pohybem venku v terénu nebo jízdě uvnitř budov. Některé podvozky jsou rychlé, ale do terénu nevhodné, jiné zase pomalé, ale s výbornou stabilitou.

Robot vnímá své okolí pomocí senzorů, proto je třeba věnovat náležitou pozornost při jejich výběru. Senzor získá reálnou informaci o stavu prostředí a převede ji na elektrický signál, který je dále zpracován řídicí elektronikou. Opět je podstatné si stanovit, jaké očekáváme prostředí a podle toho po té navrhnout vhodný senzorický systém.

Další část práce je zaměřena na správný návrh řídicí desky. Nejprve je nutné si vyjasnit, co vše bude k desce připojeno a jaké jsou požadavky na vstupy a výstupy. Případně jaké sběrnice je vhodné použít. Nesmí se zapomenout ani na indikační a příkazové prvky pro přímou obsluhu. Nakonec provedeme návrh plošného spoje.

V poslední kapitole si popíšeme, jak vytvořit firmware do mikroprocesoru, jaké budou nejlepší vývojové prostředky. Dále se zaměříme na programovou obsluhu jednotlivých senzorů a ostatních periférií. Druhá polovina této kapitoly bude věnována návrhu ovládacího programu do PC.

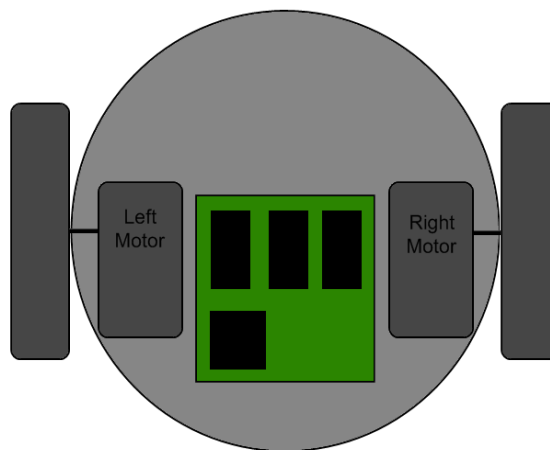
2 ROBOTICKÉ PODVOZKY

Zvolení vhodného podvozku byla první otázka, kterou jsme museli vyřešit. V této kapitole se nejdříve podíváme na různé typy podvozků používaných v robotice a provedeme srovnání z hlediska jejich parametrů. Dále prozkoumáme nabídku komerčních podvozků a nakonec vybereme správný podvozek pro zadané podmínky.

2.1 Typy robotických podvozků

2.1.1 Diferenciální podvozek

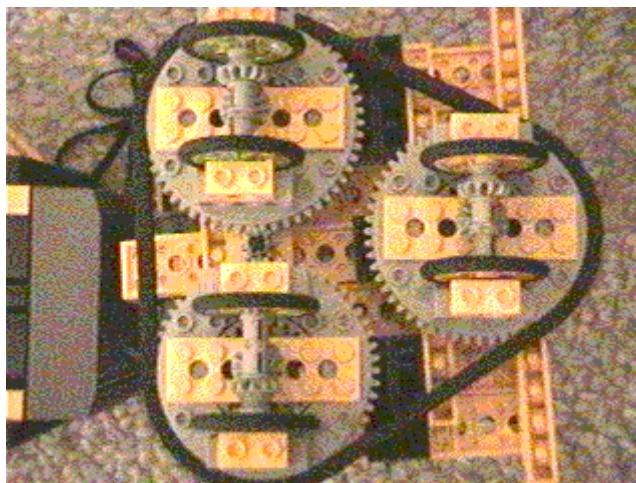
Diferenciální podvozek je charakterizován jednoduchou konstrukcí a možností snadného řízení. Výhodou také je nenáročnost na výpočet diferenciální odometrie. Naopak prostupnost terénem činí tomuto podvozku velké problémy.



Obr. 1: Diferenciální podvozek [5]

2.1.2 Synchronní podvozek

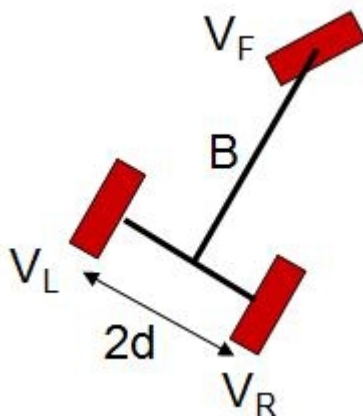
Tento podvozek se vyznačuje složitou konstrukcí a také náročnějším řízením. Zjednodušení lze provést mechanickým spřažením motorů. Většinou nabývá kruhového tvaru, aby se předešlo zachycení o překážky. Synchronní podvozek najde uplatnění i v náročnějším terénu.



Obr. 2: Synchronní podvozek [6]

2.1.3 Trojkolový podvozek s řízeným předním kolem

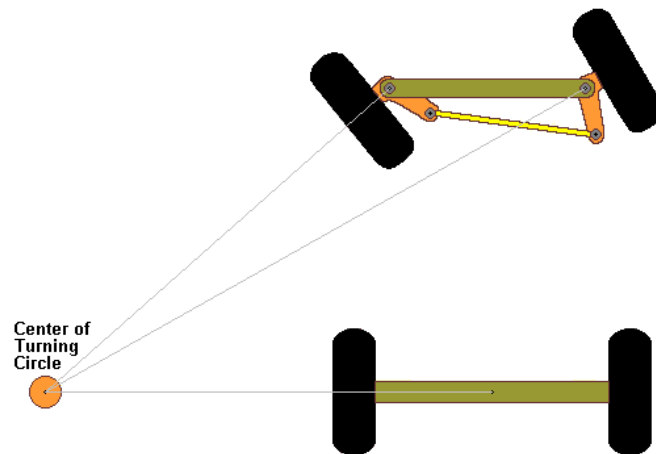
Toto řešení má oproti synchronnímu podvozku jednodušší konstrukci. Řízení je také celkem snadné, jeden motor řídí rychlost a druhý směr. Těžiště musí být uvnitř trojúhelníku mezi koly, aby byl podvozek stabilní. Je vhodný spíše do lehčího terénu.



Obr. 3: Trojkolový podvozek

2.1.4 Ackermanův podvozek

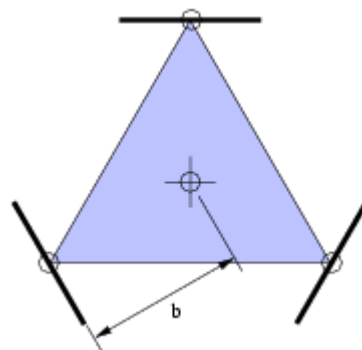
Akcermanův podvozek se používá hlavně v automobilovém průmyslu, jelikož je vhodný pro vysoké rychlosti i středně těžkém terénu. Výhodou je též malá spotřeba energie. Nevýhodou jsou holonomní omezení, díky kterým se není možno s podvozkem otočit na místě. Opustit nějaké zúžené prostory se pak může stát velikým problémem.



Obr. 4: Ackermanův podvozek [8]

2.1.5 Podvozky se všesměrovými koly

U tohoto typu je největší problém s konstrukcí kol. Dále je zde celkem složité řízení. Naopak podvozek má skvělou pohyblivost, může vyrazit libovolným směrem a ještě libovolně rotovat. Bohužel tento podvozek není vhodný do terénu.



Obr. 5: Všesměrový podvozek

2.1.6 Pásové podvozky

Podobná konstrukce i řízení jako diferenciální podvozek. Nevýhodou je větší energetická náročnost. Na druhou stranu prostupnost terénem je znatelně lepší.

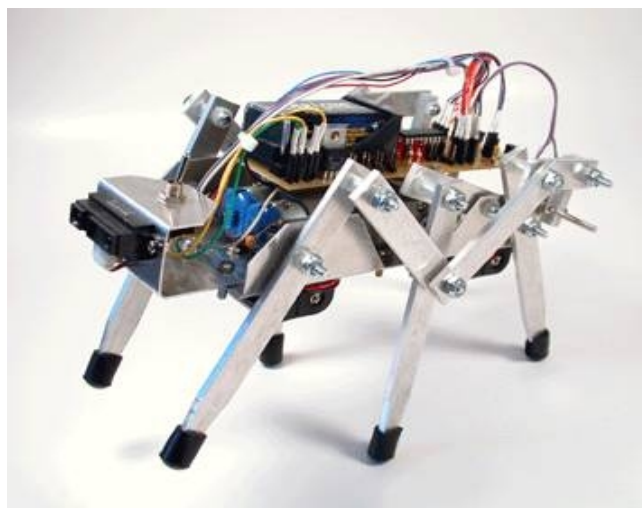


Obr. 6: Pásový podvozek [3]

Speciálním případem jsou pásové-kolové podvozky, které mají velice podobné vlastnosti jako klasické pásové. Tento typ namísto pásů používají k pohybu kola, která mohou být mechanicky nebo elektricky spřažena.

2.1.7 Kráčející podvozky

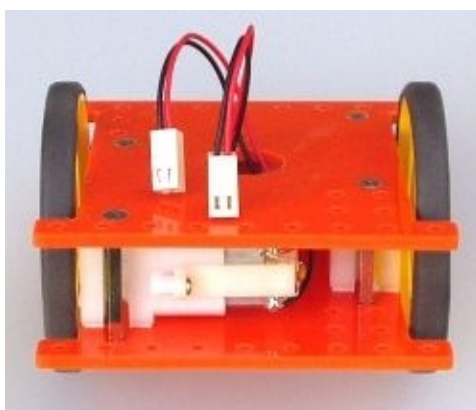
Velké množství možností v konstrukci, jak v počtu končetin, tak stupňů volnosti. Většinou se používá 6 končetin, protože toto provedení je vždy stabilní. Prostupnost terénem je výborná, ale nároky na konstrukci, řízení a stabilitu jsou hodně vysoké.



Obr. 7: Kráčející podvozek[12]

2.2 Přehled podvozků na trhu

Při výběru podvozku jsme se zaměřili pouze na český trh. Hlavním distributorem pro Českou Republiku je firma Snail Instrumets viz. [3]. Další možností jsou podvozky ze stavebnice MERKUR viz. [4]. Bohužel MERKUR se zaměřuje na robotiku příliš krátkou dobu a nabízí zatím pouze podvozky ekvivalentní podvozku UMU-01.



Obr. 8: UMU-01 [3]

UMU-01

960,-Kč

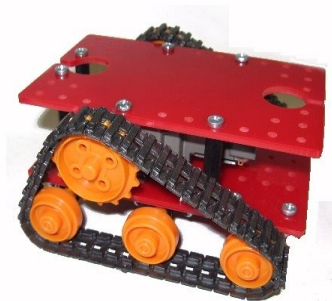
Univerzální pohonná jednotka
Základ malých dvoukolových robotů
Obsahuje 2x GM8PW

Motory jsou opatřeny odrušovacími kondenzátory, kablíky a konektory
Materiál plexi 3mm, barva červená, modrá, žlutá

Rozměry 90x90x31mm (bez kol)

Kabelová šachta průměru 18mm

Montážní díry v rastru 10mm (kompatibilní s Merkurem)



Obr. 9: TANK-01 [3]

TANK-01

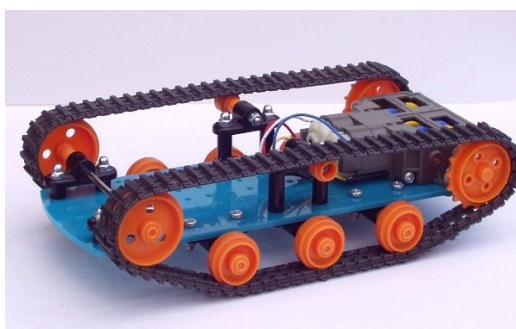
960,-Kč

Pasový podvozek pro malého robota

Obsahuje sadu pásů TM70100,
čtyřrychlostní převodovku TM70168 a
plastové díly

Rozměry 115x110x65mm

Napájení motorků 3-6V



Obr. 10: TANK-02 [3]

TANK-02

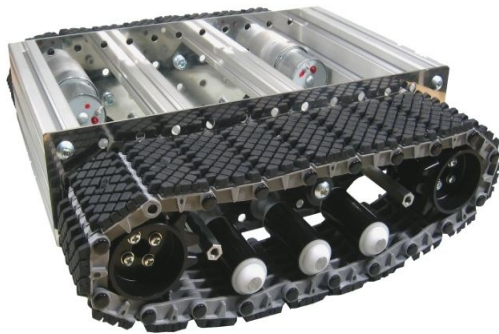
960,-Kč

Pasový podvozek pro malého robota

Obsahuje sadu pásů TM70100,
čtyřrychlostní převodovku TM70168 a
plastové díly

Rozměry 200x110x60mm

Napájení motorků 3-6V



Obr. 11: TANK-03 [3]

TANK-03

Původní cena: 7700,-Kč

Pásový podvozek

Nosnost ~5kg

Poháněno čtyřmi motory GM37, 70 ot/min

Napájecí napětí motorů 12V DC

Rozměry 260x280x95mm (d x š x v)

Materiál nerez, dural a duralové profily

Item

Použity pásy Lynxmotion TRK-01



Obr. 12: MOB-02 [3]

MOB-02

Původní cena: 3450,-Kč

Dvoukolový podvozek s pasivním opěrným otočným kolečkem

Pohon dvěma motory GM37, 120 ot/min

Průměr poháněných kol 115 mm

Průměr opěrného kolečka 50 mm

Opěrné kolečko uloženo na kuličkovém ložisku

Rozměry 265 x 270 x 115 mm (d x š x v)

Rozchod 227mm

Materiál nerez, dural a duralové profily

Item



Obr. 13: MOB-03 [3]

MOB-03

Původní cena: 4900,-Kč

Čtyřkolový podvozek s pohonem všech kol (4x4)

Nosnost ~5kg

Poháněno čtyřmi motory GM37, 120 ot./min

Průměr poháněných kol 115 mm

Rozměry 240 x 270 x 115mm (d x š x v)

Rozchod 227mm Materiál nerez, dural a duralové profily

Item

2.3 Výběr podvozku

Podle zadání vybíráme podvozek pro pohyb v průmyslovém prostředí. To znamená podvozek, který bude schopen překonat základní nerovnosti jako práh u dveří, mírné stoupání nebo klesání a další.

Dalším požadavkem je rychlost pohybu robotu. Pokud bude převážet něco nebezpečného je nežádoucí, aby se pohyboval v pracovním prostoru dlouhou dobu.

Jelikož předpokládáme použití jednoho mikropočítače, tak by bylo vhodné použít podvozek s menšími nároky na řízení.

Na základě průchodnosti terénem jsme vyřadili diferenciální, trojkolový, všesměrový a Ackermanův podvozek, který se musí složitě otáčet na místě. Z hlediska rychlosti se jeví jako nevhodný krácející typ podvozku a synchronní podvozek nevyhovuje zase našim požadavkům na řízení.

Nejvhodnější typ pro naše účely bude pásový podvozek. Jelikož předmětem práce není návrh podvozku, ale pouze jeho výběr, volíme jeden z komerčních podvozků. Na výběr tedy zůstal TANK-03 a MOB-03. Oba mají přibližně stejné rozměry i hmotnost. Volíme podvozek MOB-03, jelikož s ním lze dosáhnout větší rychlosti a je také znatelně levnější. Vybraný typ se nachází na obr.13.

3 SENZORY PROSTŘEDÍ

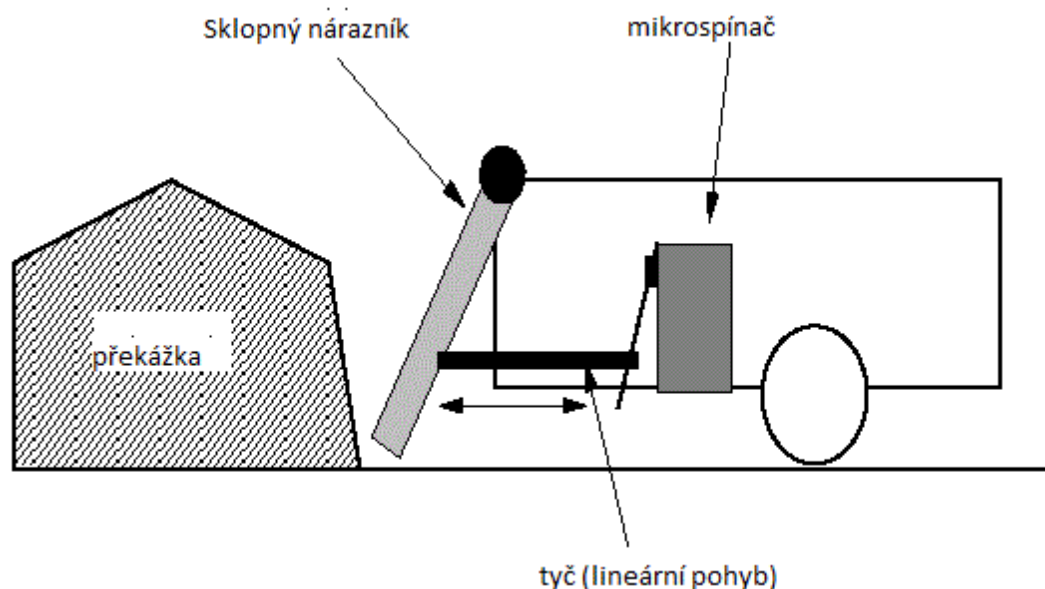
Po výběru vhodného podvozku, je dalším úkolem vybrat správné senzory. Ty slouží k mapování překážek v okolí robota. Pomocí nich se tedy snažíme určit polohu, tvar, barvu a další vlastnosti překážek.

Nejprve se podíváme na různé senzory obecně, jaké jsou typy a co s nimi lze naměřit. Potom opět zjistíme, co lze zakoupit za komerční produkty a nakonec navrhne sensorický systém pro náš robot.

3.1 Druhy senzorů

3.1.1 Taktilní senzory

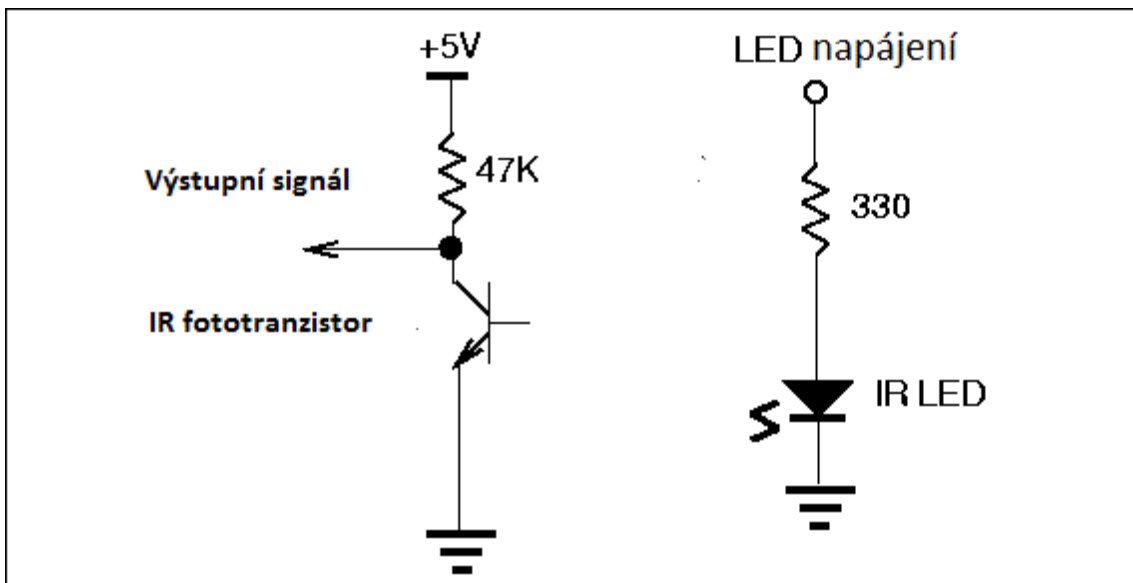
Jsou nejzákladnějším typem senzoru. Většinou jsou realizovány kontaktním spínačem. Při interakci senzoru s překážkou dojde ke změně logické úrovně na výstupu čidla. Senzor lze tedy přímo připojit na vstup mikropočítače, který může pomocí přerušení okamžitě reagovat na kontakt s překážkou. Mezi taktilní senzory také patří tenzometry a senzory založené na piezoelektrickém jevu. Taktilních senzorů se využívá tam, kde může dojít ke kontaktu s překážkou či jiným objektem. Bohužel tyto snímače mají pouze omezený počet sepnutí, což je dáno jejich konstrukčními vlastnostmi. Dále je problém s tím, že výstup je aktivní až při nárazu do překážky, proto se lze často setkat hlavně s mechanickou poruchou tohoto senzoru.



Obr. 14: Taktilní senzor [7]

3.1.2 Infračervené senzory reflexní

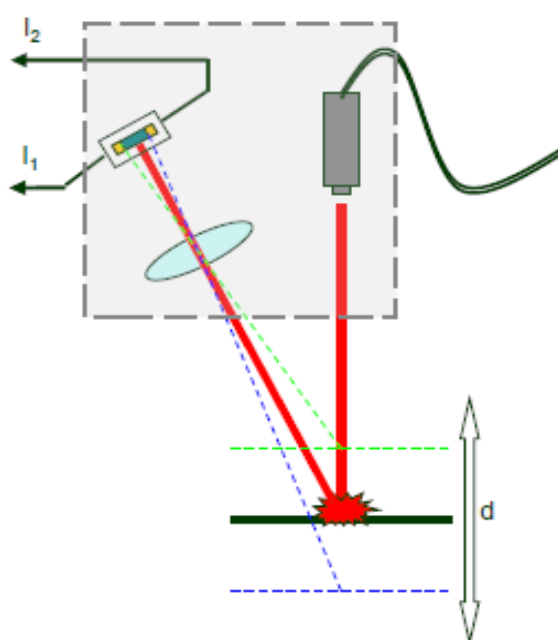
Infračervené detektory překážek slouží k detekování objektů v blízkém okolí robota, řádově desítky centimetrů. Principem IR detektoru překážek je detekce odraženého infračerveného světla od překážky. Jako zdroj světla je použita infračervená LED dioda a jako detektor fototranzistor citlivý v IR oblasti. Tento senzor opět poskytuje logický signál o tom, zda se před ním nachází nebo nenachází překážka. Nevýhodou těchto IR senzorů pracujících na principu detekce odraženého IR světla je, že množství odraženého světla je závislé na barvě překážky a druhu povrchu. Toto lze eliminovat nahrazením fototranzistoru specializovaným IR přijímačem s integrovaným demodulátorem. Jeho výhodou je, že je citlivý pouze na modulované IR záření o určité vlnové délce generované IR LED diodou. Modulační kmitočty bývají 36-56 kHz. Důvodem použití modulace je eliminování vlivu IR záření v okolním světě.



Obr. 15: Infračervený senzor [7]

3.1.3 Odrazové triangulační infra senzory

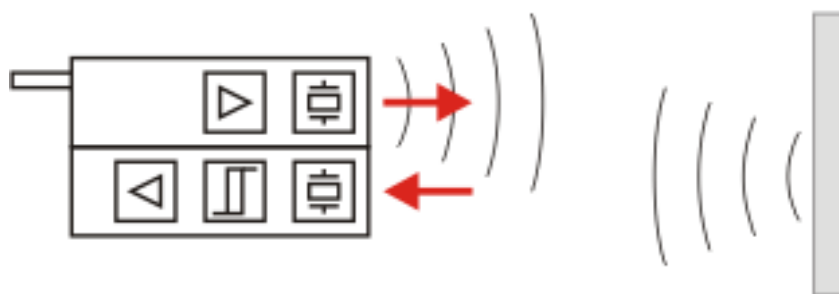
Tento typ senzoru je vybaven zdrojem světla, který vysílá obrazec pod definovaným úhlem, většinou kolmo na detekovaný objekt. Snímač zjistí úhel dopadu odraženého paprsku a jeho základě vypočítá vzdálenost překážky. Pro vyhodnocení úhlu dopadu se používá polohově citlivý snímač, který využívá fotoliteární jev. Generovaný paprsek dopadá na velkoplošnou fotodiodu PIN, kde se v I vrstvě generují páry elektron-díra, ovlivňované elektrickým polem. Celkový proud se dělí na dvě složky, ze kterých lze určit pozici dopadu paprsku.



Obr. 16: Odrazový triangulační infrasezor [9]

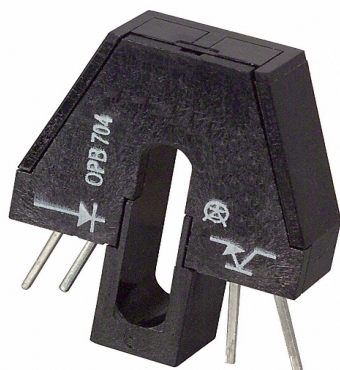
3.1.4 Ultrazvukové senzory

Princip měření vzdálenosti k překážce je založen na principu měření doby mezi vysláním akustického signálu a přijetím odraženého akustického signálu – echa. Ultrazvukové senzory pracují na kmitočtu vyšším než je schopno lidské ucho zaznamenat. Nejběžnější frekvence jsou nad 40 kHz. Díky relativně nízké rychlosti zvuku (ve vzduchu) je doba mezi vysláním a přijmem signálu výrazně vyšší než u radarových, laserových a také IR senzorů. Proto lze dosáhnout relativně vysoké přesnosti měření i bez extrémních nároků na vyhodnocovací obvody. Vzdálenost objektu je možno vypočítat pomocí známé rychlosti šíření akustických signálů ve známém prostředí měřením času, který uplyne od okamžiku vyslání signálu až do příjmu odražené vlny. Nejsilnější signál bývá odražen od překážek, které jsou pevné, hladké a ploché a směřující kolmo k vysílanému svazku. Nevýhodou je perioda měření vyšší než 0,1 s a tlumení ultrazvukového signálu, což omezuje dosah přibližně na 10 m. Důležitým faktorem je také vliv teploty vzduchu na rychlost šíření zvuku. Ta je pro teplotu $t = 0\text{ °C}$; $v = 331\text{ m/s}$ a pro $t = 25\text{ °C}$; $v = 343\text{ m/s}$.



Obr. 17: Ultrazvukový senzor[10]

3.2 Přehled komerčních senzorů



Obr. 18: OPB704G [3]

OPB704B

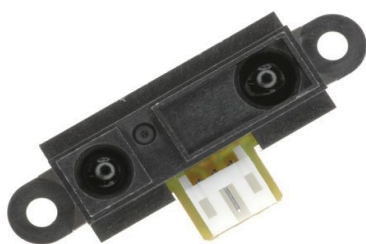
85,-Kč
Odrazový infrasenzor
IR LED + fototranzistor
Infračervený filtr
Detekce na vzdálenost <15mm,
optimum 3-10mm
Vhodný pro sledování čáry, detekci
překážek či mantinelů
Upevnění šroubem M3 s podložkou



Obr. 19: QTR-1A [3]

QTR-1A

71,-Kč
QTR-1A Reflectance Sensor
Modul s odrazovým infrasenzorem
IR LED + fototranzistor
Výstup analogový
Detekce na vzdálenost <9mm,
optimum 3mm
Vhodný pro sledování čáry či detekci okraje
arény



Obr. 20: GP2Y0A21 [3]

GP2Y0A21

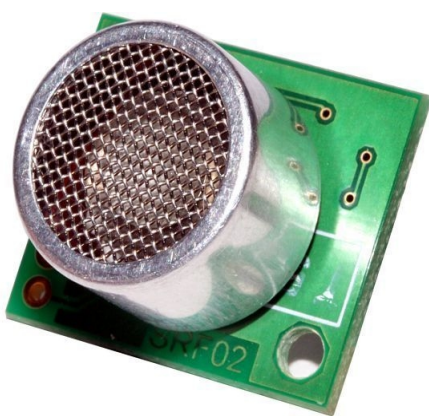
370,-Kč
Odrazový infrasenzor
Novější náhrada za GP2D12 a GP2D02
Detekce na vzdálenost 10-80cm
Vhodný k detekci soupeře či překážek
Rozměry 40x13x13.5mm (podrobněji v kat.
listu)
Napájení 5V/30mA
Výstup je napětí úměrné vzdálenosti
Součástí je i protikus konektoru



Obr. 21: GP2Y0D805 [3]

GP2Y0D805

Odrazový infrasenzor 5cm
Dvoustavový výstup
Rozměry 14x7x8mm (podrobněji v kat. listu)
Napájení 2.7-6.2V/9mA
Montáž do plošného spoje, rozteč vývodů 1.5mm



Obr. 22: SRF02 [3]

SRF02

Ultrazvukový dálkoměr
S jedním měničem
Frekvence 40kHz
Vyzařovací úhel 55°
Napájení 5V/4mA typ.
Rozsah 16cm - 6m
Analogové zesílení automatické v 64 krocích
Výstup I2C interface a asynchronní sériový interface
Možnost adresovat až 16 zařízení na obou typech interface
Možnost vyslání pulsu bez měření a naopak měření bez vyslání pulsu



Obr. 23: SRF05 [3]

SRF05

Ultrazvukový dálkoměr
Zpětně kompatibilní s dřívějším SRF04
Frekvence 40 kHz
Vyzařovací úhel 55°
Napájení 5V/30mA
Rozsah 1cm - 4m
Startovací puls min. 10us, TTL
Výstup impuls délky 0.1 - 25ms

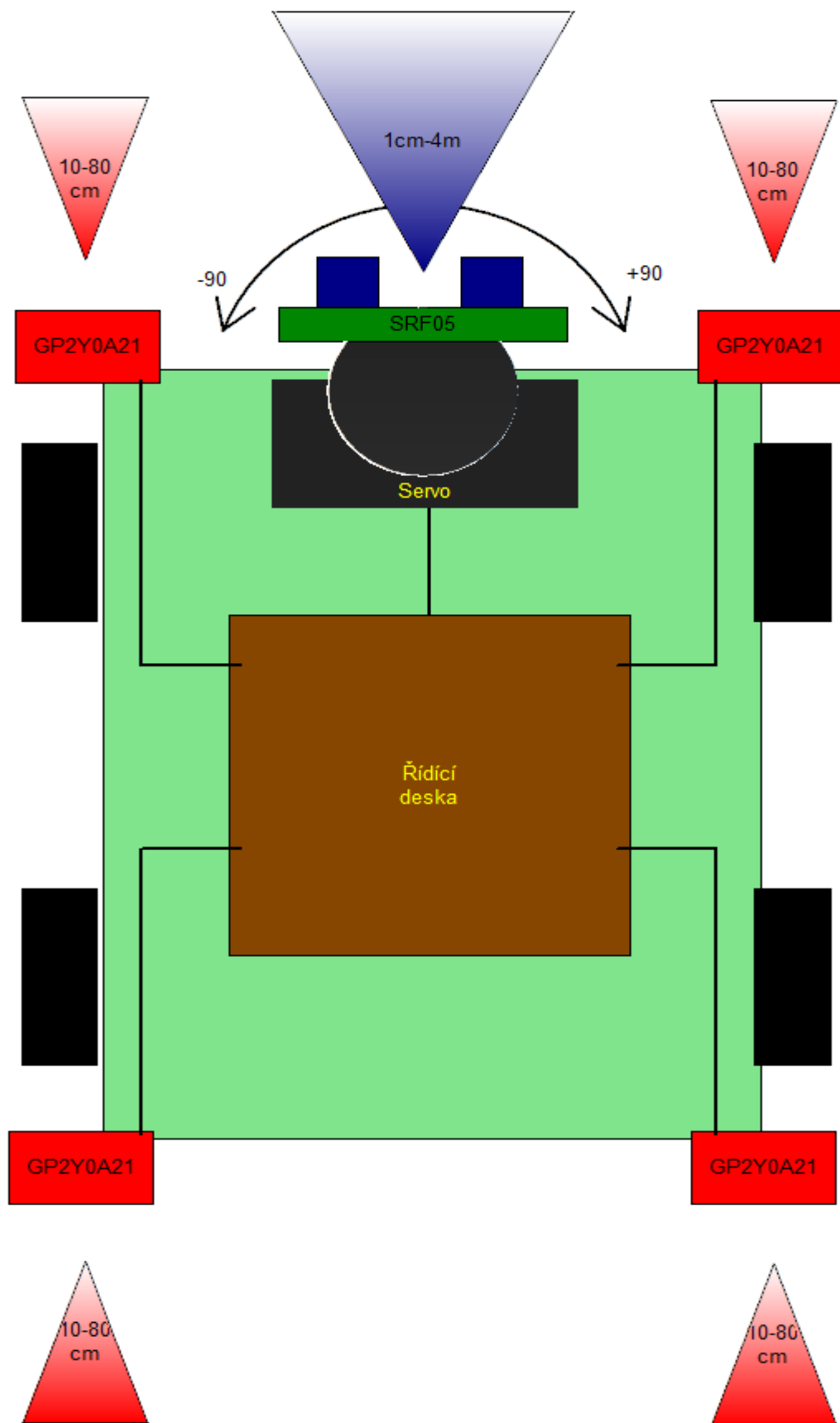
3.3 Návrh senzorického systému

Pokusíme se navrhnout vhodný senzorický systém pro pohyb v průmyslovém prostředí. Za průmyslové prostředí považujeme výrobní haly, testovací místnosti nebo podobné prostory. Předpokládáme pohyb po rovném povrchu a jako hlavní druh překážek očekáváme zdi. Dále by měl být robot schopen rozpoznat objekty rozestavené po místnosti jako například tašku s nářadím apod.. Nepředpokládáme umístění polopropustných překážek, jako je například drátěný plot nebo různé sítě.

Důležité je, aby robot do žádné z předpokládaných překážek nenaboural a to ani při couvání. Pro správný a plynulý pohyb musí mít robot přehled i o tom co se nachází dál ve směru předpokládaného pohybu.

Použití taktilních senzorů jsme rovnou vyloučili kvůli jejich nespolehlivosti a nutnosti interakce s předmětem. Infračervené senzory OPB704B a QTR-1A jsou nevhodné z důvodu malého rozsahu, lze detekovat objekt pouze do vzdálenosti 1 cm. GP2Y0D805 detekuje předměty až do 5 cm, ale i to by mohlo být málo při větších rychlostech, navíc jeho výstup je pouze binární. Nejvhodnější typ pro naše použití je GP2Y0A21. Jeho rozsah začíná sice až na 10 cm, což ale v našem případě stačí. Jeho výstup je analogový a velikost napětí je úměrná vzdálenosti od překážky. Kombinací s vhodným softwarem lze realizovat i funkci koncových spínačů. Proto jsme se rozhodli použít tento senzor nad každé kolo, čímž by mělo být zajištěno, že robot nenabourá.

Pro sledování vzdálenějších objektů je vhodné použít ultrazvukové čidlo. Model SRF02 obsahuje pouze jednu ultrazvukovou vložku a proto jeho rozsah začíná až na 16 cm, což by někdy mohlo být problematické. Typ SRF05 už má implementovány dvě ultrazvukové vložky jeho rozsah je 1 cm až 4 m. Výstup je binární na úrovních TTL přičemž délka pulzu je úměrná vzdálenosti. Stejně vlastnosti má i SRF06, akorát výstup je analogový 4-20 mA, což je pro přímé měření procesorem nevhodné, jelikož bychom museli použít speciální A/D převodník. Dále je možné sehnat ještě modely SRF08 a SRF10, které mají navíc ještě implementovány I²C sběrnice pro snadný přístup, bohužel za téměř dvojnásobnou cenu. Rozhodli jsme se použít jeden senzor SRF05 umístěný na servomotoru, jelikož v měření délky pulzu mikroprocesorem nevidíme až takový problém. Navíc v kombinaci se servomotorem lze sledovat široké okolí před robotem s dostatečným rozsahem. Navržený senzorický systém je zobrazen na obr. 24.



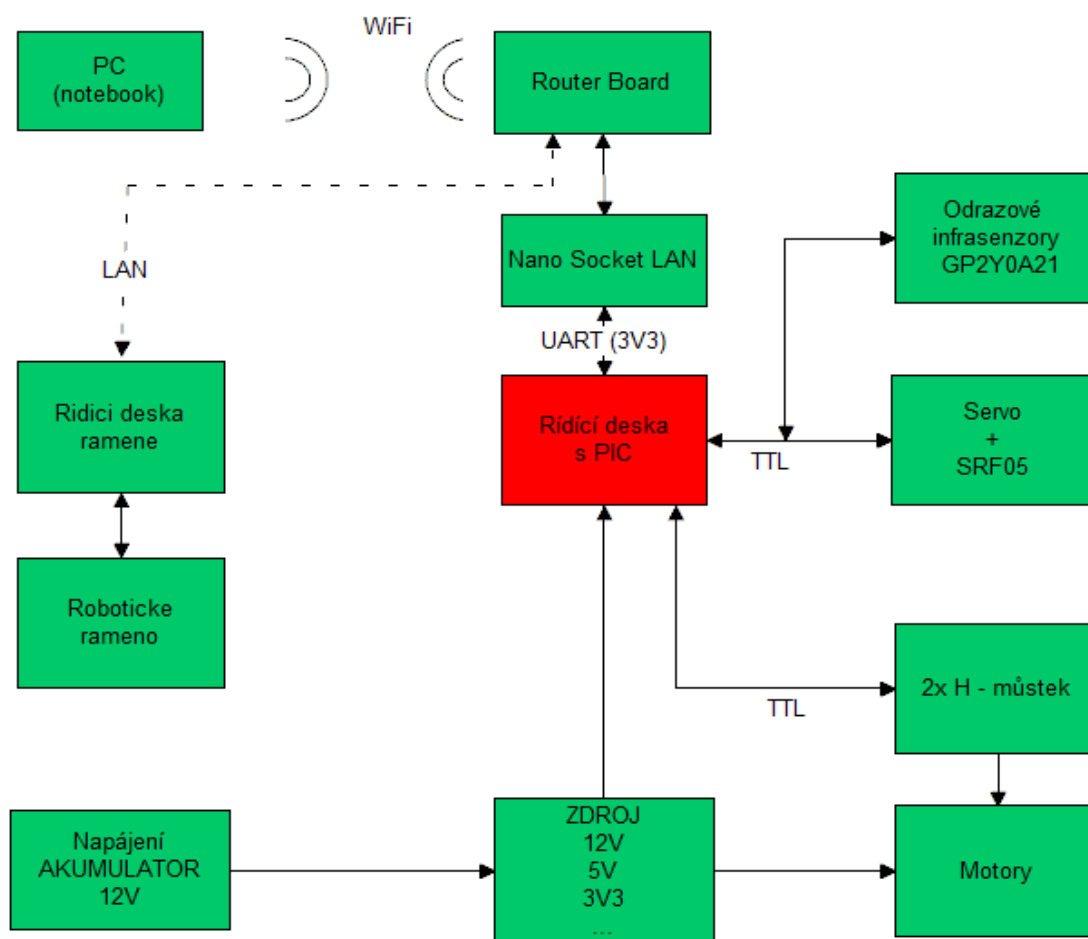
Obr. 24: Blokové schéma navrženého senzorického systému

4 ŘÍDÍCÍ DESKA

4.1 Požadavky na řídicí jednotku

Blokové schéma celého robotu je zobrazeno na obr. 25. Vybraný podvozek obsahuje čtyři stejnosměrné motory u nichž předpokládáme možnost otáčení oběma směry a pokud to bude možné tak i plynulou regulaci rychlosti. Každé dva motory budou spojeny paralelně a pro jejich činnost budou použity výkonové H-můstky.

Pro obsluhu infračervených senzorů GP2Y0A21 jsou nutné čtyři analogové vstupy, které budou zpracovávat výstupní napětí ze senzorů.



Obr. 25: Blokové schéma robotu

Dále musíme zajistit obsluhu ultrazvukového sonaru a servomotoru, který s ním bude otáčet. Senzor SRF05 se aktivuje jedním pulzem na vstupu a čeká se na echo na výstupu. Proto budeme potřebovat jeden digitální výstup pro generování pulzu a jeden

digitální vstup pro indikování echa. Na otáčení servomotoru použijeme ještě jeden digitální výstup. Komunikace se vzdáleným operátorem bude probíhat po sériové lince RS232 na 3,3 V logice, proto náš procesor musí tuto linku podporovat.

4.2 Zvolený procesor

Pro řízení našeho robota jsme se rozhodli použít procesor od firmy Microchip PIC16F877A, který vyhovuje všem našim požadavkům. Dalším důvodem jsou naše zkušenosti s aplikací tohoto typu procesoru. Přehled základních vlastností najdeme níže v tabulce 1, další detaily lze získat z online datasheetu [11].

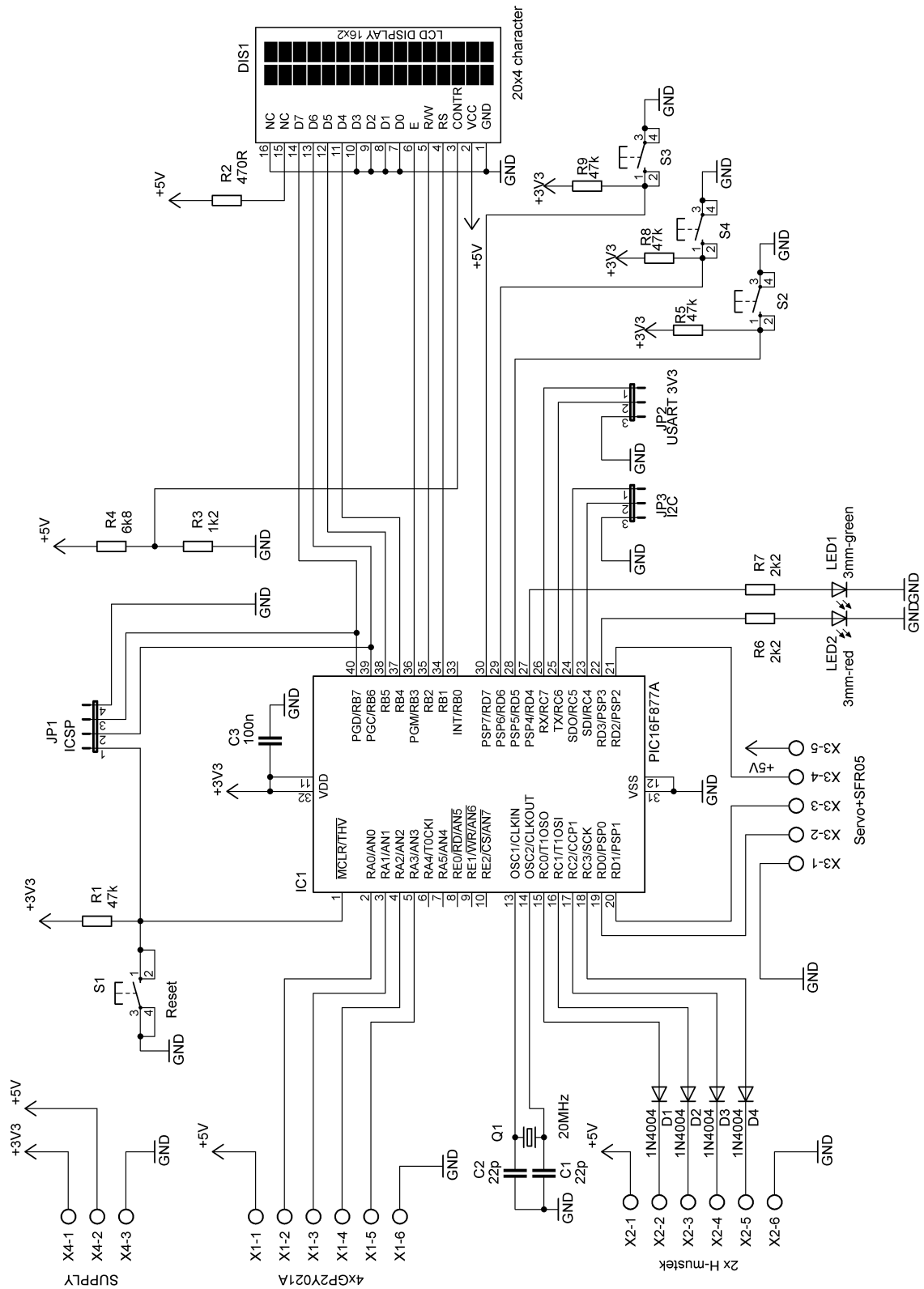
Program Memory Type	Flash
Program Memory (KB)	14
CPU Speed (MIPS)	5
RAM Bytes	368
Data EEPROM (bytes)	256
Comunication Peripherals	1-A/E/USART, 1-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	2 CCP
Timers	2 x 8-bit, 1 x 16-bit
ADC	8 ch, 10-bit
Comparators	2
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	2 to 5.5
PWM	2 x 10-bit
EEPROM	256 Bytes

Tabulka 1: Základní parametry PIC16F877A [11]



Obr. 26: Mikroprocesor PIC16F877A

4.3 Schéma řídicí desky



Obr. 27: Schéma řídicí desky

Základem řídicího systému je procesor PIC16F877, který provádí obsluhu veškerých periférií, které jsou připojeny přímo na analogové/digitální vstupy nebo výstupy. Dále jsou na řídicí desce umístěny tlačítka a LCD displej pro snadnější odladění aplikace. Nano Socket LAN modul je připojen přímo vodiči Rx.Tx a GND. Deska je napájena 3,3 V a displej 5 V. Obvody pro stabilizaci jednotlivých napětí jsou na samostatné desce.

4.4 Seznam součástek

Rezistory :

R1,R5,R8,R9	47k	0,25W, 5%
R2	470R	0,25W, 5%
R6,R7	2k2	0,25W, 5%
R3	1k2	0,25W, 5%
R4	6k8	0,25W, 5%

Kondenzátory:

C1,C2	22p	keramický, 5mm, 63V
C3	100n	keramický, 5mm, 63V

Diody:

D1-D4	1N4004	1A,80V
-------	--------	--------

LED Diody:

LED1	zelena-3mm
LED2	cervena-3mm

Integrované obvody:

IO1	PIC16F877A	jednočipový mikropočítač, pouzdro DIL40
-----	------------	--

Konektory:

JP2,JP3	JP2E	3 x jumper
JP1	JP4E	4 x jumper
X1,X2	AK300/6	6-pin svorkovnice
X3	AK300/5	5-pin svorkovnice
X4	AK300/3	3-pin svorkovnice

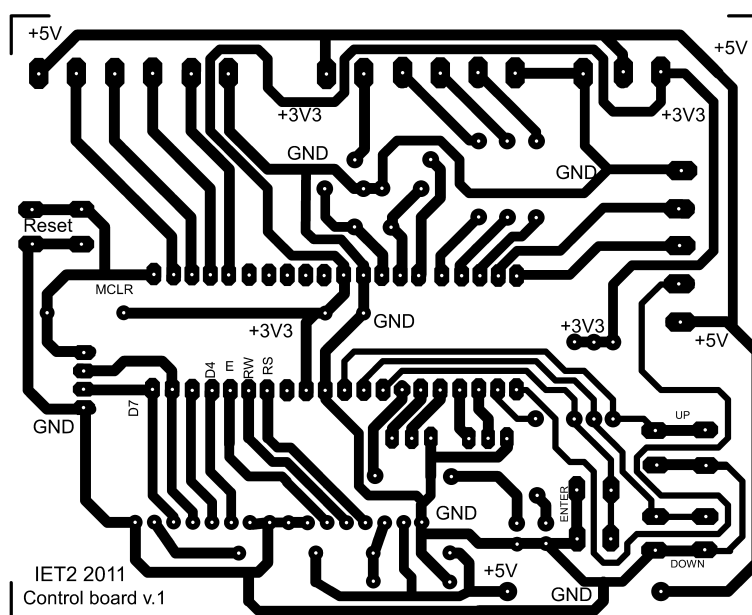
Krystaly:

Q1	Q20Mhz	20MHz
----	--------	-------

4.5 Deska plošných spojů

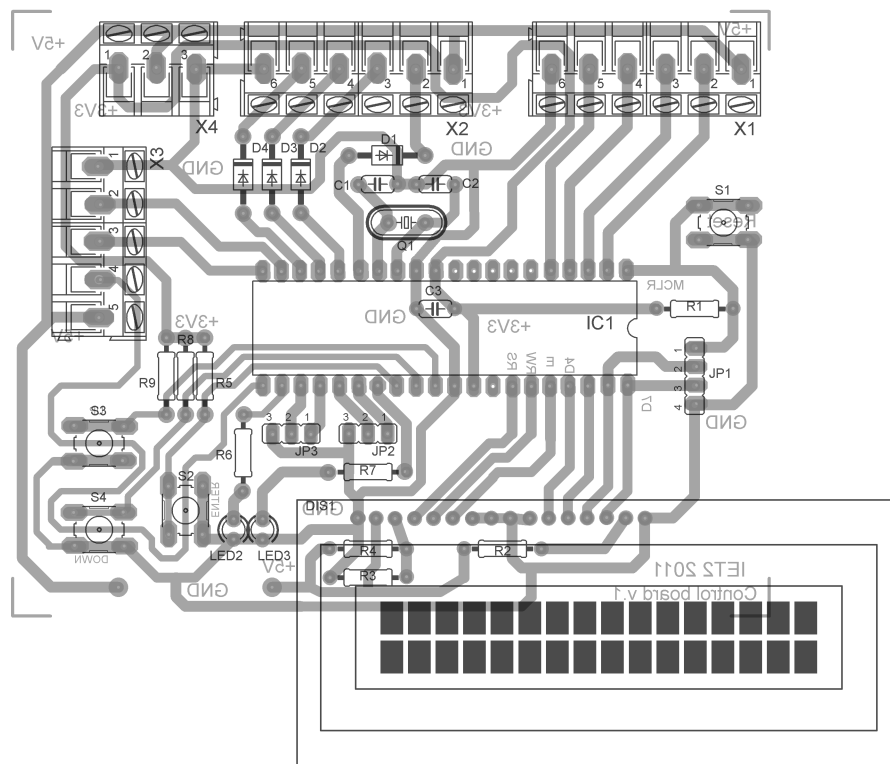
Byla navržena jednostranná deska plošných spojů v programu Eagle. Zvolili jsme klasické provedení, jelikož lze snadněji provádět diagnostiku a optimalizaci aplikace. Pro připojení napájení, senzorů a ostatních periférií jsou použity standardní svorkovnice. Desku tedy můžeme v případě potřeby snadno demontovat samostatně bez ostatních připojených prvků. V budoucnu se předpokládá návrh nové desky, pravděpodobně v SMD provedení.

4.5.1 Motiv plošného spoje



Obr. 28: Motiv plošného spoje

4.5.2 Osazovací výkres



Obr. 29: Osazovací výkres

5 PROGRAMOVÉ ŘEŠENÍ

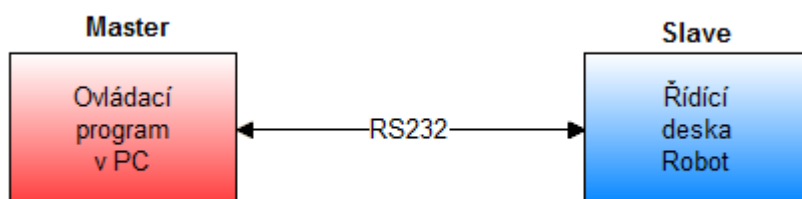
5.1 Celková koncepce

Ovládací program v PC a řídicí deska na robotu spolu komunikují pomocí klasické sériové linky. Na řídicí desce je reprezentována jako 5V TTL logika, kterou dále zpracovává NanoSocketLAN modul a ten už pak přímo připojen na Ethernet. V PC je tato linka emulována přes UDP protokol, kde běží na specifickém portu. Potom se to jeví jako, že robot má svou vlastní IP adresu. Parametry konfigurace jsou uvedeny v tabulce níže. Samotná bezpečnost a správnost komunikace je ošetřena v nižších vrstvách přenosu, ale to není předmětem této práce.

Baud Rate [Bd]	57600
Stop bitů [-]	1
Parita [-]	Žádná
Datových bitů [-]	8
RTS [-]	Ne
DTR [-]	Ne

Tabulka 2: Konfigurace RS232

Pro vzájemnou komunikaci bylo nutné do jednotlivých programů implementovat tvar a způsob jednotlivých zpráv. Topologie tohoto spojení je typu master - slave. Za master považujeme ovládací program v PC a jako slave je brán samotný robot.



Obr. 30: Zjednodušený pohled na komunikaci

5.2 Druhy zpráv

Master vždy vysílá pouze jeden byte, který jednoznačně určuje o jaký se jedná povel. Toto řešení bylo zvoleno z důvodu snadného dekodování příkazu robotem. V programu poté stačí implementovat jednoduchý přepínač, kde jednotlivé stavy jsou určeny daným povellem.

Program=>Robot			Význam
Hex	Dec	ASCII	
0x30	48	0	Autonomní jízda
0x31	49	1	STOP
0x61	97	A	Jízda vpřed
0x62	98	B	Jízda vzad
0x63	99	C	Otoč vlevo
0x64	100	D	Otoč vpravo
0x65	101	E	Sonar pravá
0x66	102	F	Sonar levá
0x67	103	G	Snímaní Zap.

Tabulka 3: Přehled povelů pro robot

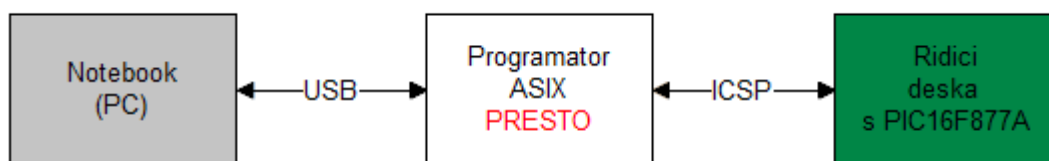
Slave naopak vysílá delší zprávy, každá začíná start bytem, který je reprezentován jako znaménko "#". Za ním je odesláno dalších 8 bytů, přičemž každý z nich jednoznačně určuje vzdálenost objektu od daného senzoru. Hodnoty jsou přímo čísla od 0 do 255. Samotný převod na centimetry provádí program v PC.

Byte[]	Robot=>Program	
0	0x23 "#"	Start byte
1	Pravý přední PP	0x00 až 0xFF
2	Levý přední LP	0x00 až 0xFF
3	Pravý zadní PZ	0x00 až 0xFF
4	Levý zadní LZ	0x00 až 0xFF
5	Sonar P2	0x00 až 0xFF
6	Sonar P1	0x00 až 0xFF
7	Sonar S	0x00 až 0xFF
8	Sonar L1	0x00 až 0xFF
9	Sonar L2	0x00 až 0xFF

Tabulka 4: Zpráva od robota

5.3 Firmware PIC16F877A

Firmware do mikroprocesoru PIC16F877A je napsán v jazyku C. Jako vývojové prostředí jsme zvolili mikroC for PIC. Pro nahrávání programu do procesoru byl použit programátor ASIX PRESTO, který se ovládá pomocí programu ASIX UP. Propojení desky s programátorem je zajištěno přes ICSP rozhraní.



Obr. 31: Způsob programování PIC16F877A

5.3.1 Vývojové prostředí mikroC for PIC

Vydavatelem tohoto prostředí je firma mikroElektronika [15], která se zabývá výrobou a distribucí různých vývojových platforem pro neznámější typy procesorů jako jsou AVR, PIC, 8051 a další.

Tento nástroj má ve své základní výbavě implementováno spousty knihoven pro práci s různými periferiemi. Najdeme zde například knihovnu “Lcd Library“ pro práci s LCD displejí nebo “Acd Library“ sloužící k obsluze A/D převodníků a mnoha dalších. Podpora těchto knihoven nám velice usnadňuje práci, nemusíme se zabývat tím jak na displej dostat data, prostě jenom používáme již vytvořené funkce.

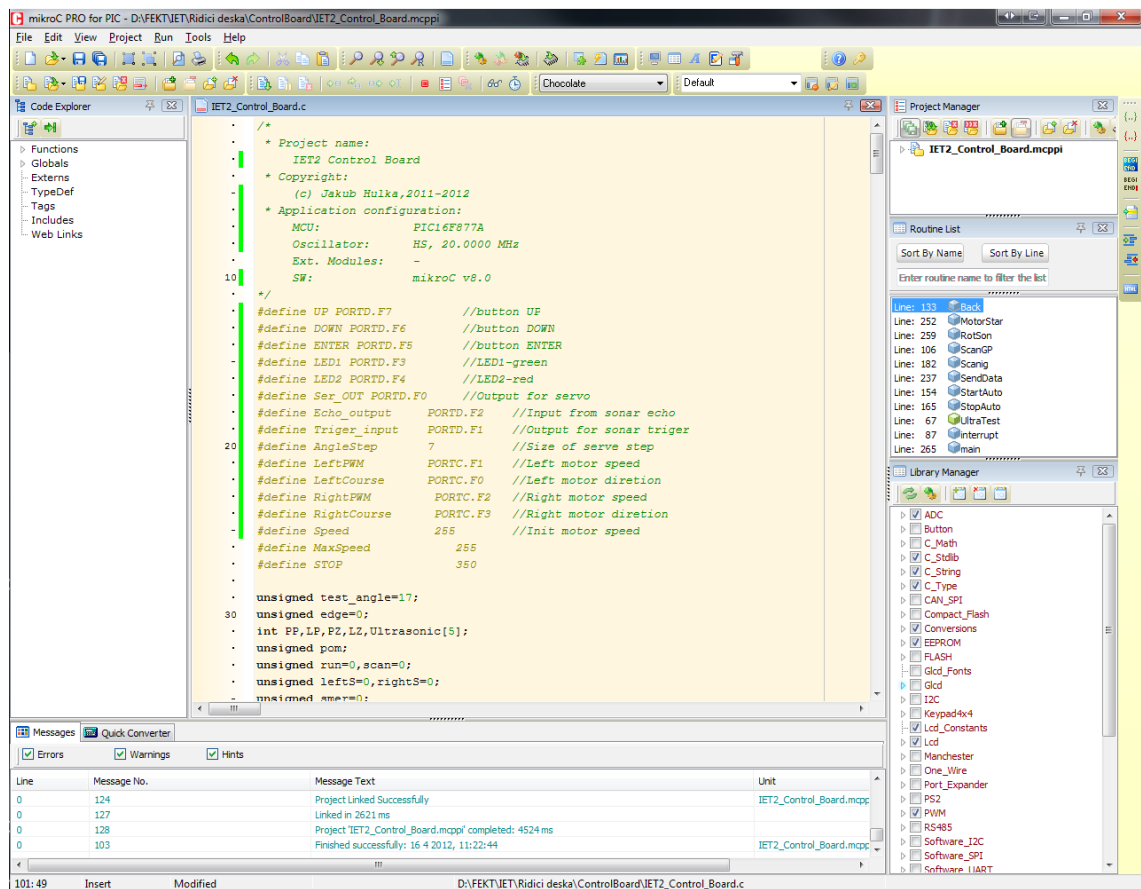
Další výhodou tohoto programu je implementace různých utilit pro samotné odladění aplikace. Najdeme zde různé druhy terminálů USART terminal, UDP terminal a jiné. Utilitu pro tvorbu vlastních LCD znaků a ostatní užitečné věci.

V prostředí lze také snadno získat různé statistiky o velikosti použité paměti, o četnosti výskytu různé funkce nebo kde se v paměti nachází konkrétní proměnná.

V levé části programu najdeme nastavení projektu, kde lze vybrat typ procesoru a zadat jeho taktovací frekvenci, od které se pak dynamicky dopočítávají hodnoty

různých zpoždění. Vpravo máme výpis všech použitých funkcí a seznam implementovaných knihoven. Editace samotného kódu probíhá klasicky uprostřed prostředí. Dolní část programu je vyhrazena pro informační a chybové hlášení, které nastanou během kompilace.

V našem případě byla použita demo verze 2.50, aktuální verzi lze stáhnout zde: <http://www.mikroe.com/eng/products/view/7/mikroc-pro-for-pic/> . Tato verze je omezena pouze maximální velikostí výstupního hexa kódu na 2kB paměti.



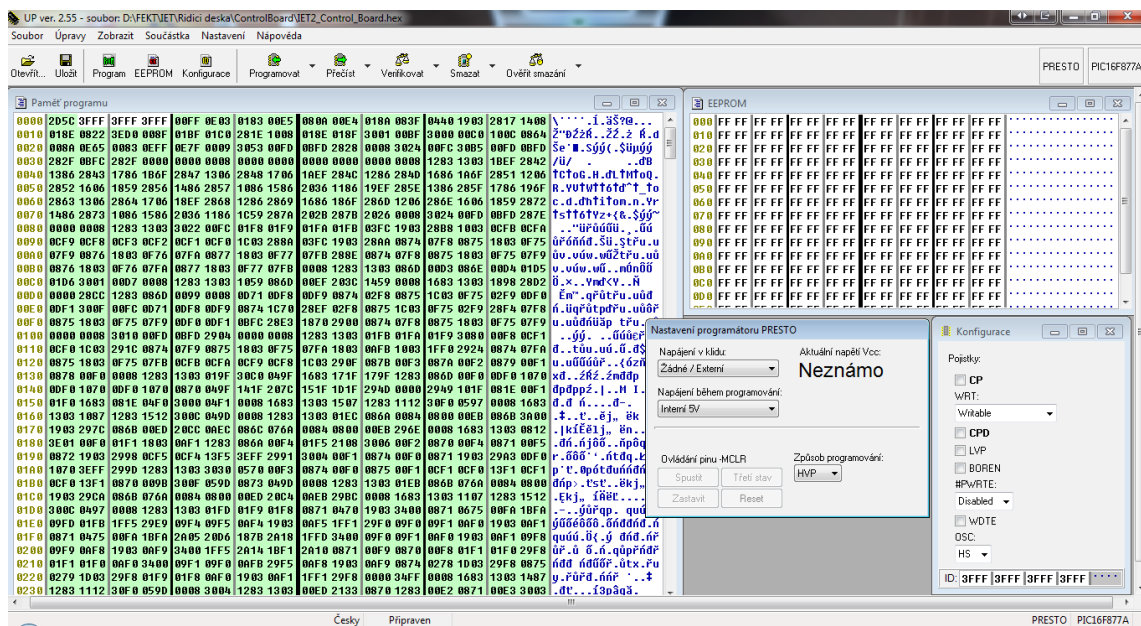
Obr. 32: Ukázka prostředí MikroC for PIC

5.3.2 Program ASIX UP

Tento program si můžeme volně stáhnout přímo ze stránek výrobce firmy ASIX s.r.o <http://asix.cz/index.htm> . Jedná se o českou společnost a tak není problém ani s případnou podporou. Uživatelské prostředí poskytuje snadné, intuitivní a rychlé ovládání všech funkcí programátoru. Kromě všech standardních funkcí (načtení

souborů, programování, verifikace, vyčtení součástky, ...) je zde i řada dalších vylepšení, ty nejzajímavější jsou:

- Editace (včetně podpůrných funkcí) a zpětné uložení programovaných dat
- Barevné rozlišení dat podle původu (součástka, soubor, ruční editace...) a podle výsledku verifikace (nesouhlasící hodnoty jsou zvýrazněny)
- Prostředí v maximální míře respektuje specifické vlastnosti PIC:
- Formát zobrazovaných dat je optimalizován podle šířky instrukčního slova daného PIC
- Možnost zobrazení a nastavení jednotlivých pojistek i přímá editace ve formátu konfiguračního slova
- Je možné programovat i verifikovat zvlášť paměť programu, datovou paměť EEPROM a konfigurační slovo a ID.
- Při použití vhodného programátoru (např. PRESTO) lze po ICSP naprogramování přímo aplikaci spustit (softwarové ovládání Stop/Run/Reset)
- Ukončení programování a jeho výsledek je volitelně doprovázeno nastavitelnou zvukovou signalizací.
- Software automaticky ukládá poslední nastavení - typ programované součástky, jméno souboru * .HEX (včetně celé cesty) a všechny potřebné parametry.



Obr. 33: Ukázka programu ASIX UP

5.3.3 Programátor PRESTO

PRESTO je velmi rychlý programátor moderní koncepce. Podporuje programování osazených součástek - ISP (In-System Programming). Programuje velké množství součástek - mikrokontroléry, CPLD, FPGA, sériové FLASH a EEPROM, apod. řady významných výrobců. S počítačem je propojen rozhraním USB.

PRESTO programuje mikrokontroléry Microchip PIC a vybrané dsPIC, Atmel AVR a 8051, 32-bitové mikrokontroléry s jádrem ARM (Atmel, Philips/NXP, ...) a další. Také zvládá programování různých pamětí Flash a EEPROM.

Základní vlastnosti:

- Velmi rychlé provádění operací
- Řízeno a napájeno z USB
- Napájení cílové aplikace od 3 do 5 V
- Ochrana proti přepětí a nadproudu
- Tlačítko GO
- Pokročilý software (pro Windows i Linux)

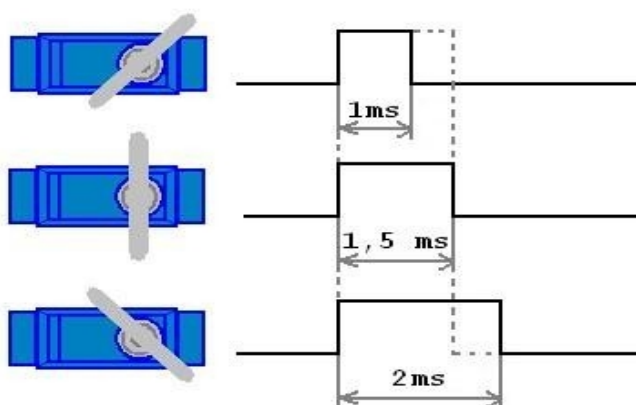


Obr. 34: Programátor PRESTO

5.3.4 Obsluha serva+SRF05

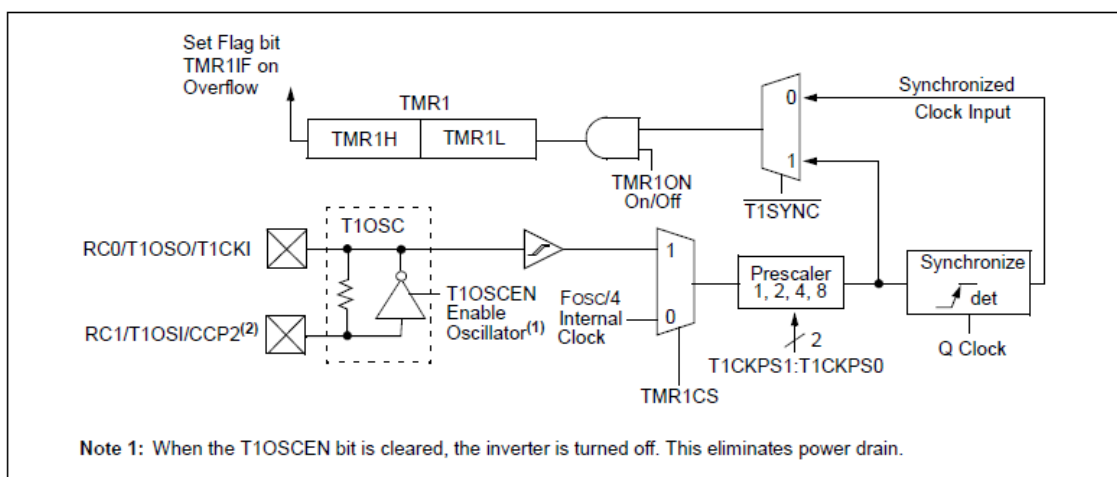
Pro získání přehledu o stavu překážek před robotem jsme zvolili kombinaci ultrazvukového sonaru SRF05 v kombinaci s klasickým modelářským servomotorem HS-422.

Poloha servomotoru se řídí pulsem o délce 0,5 až 2,5 ms, jak můžeme vidět na obr. 35. Puls délky 0,5 ms odpovídá jedné krajní poloze a 2,5 ms puls poloze o 180° dále.



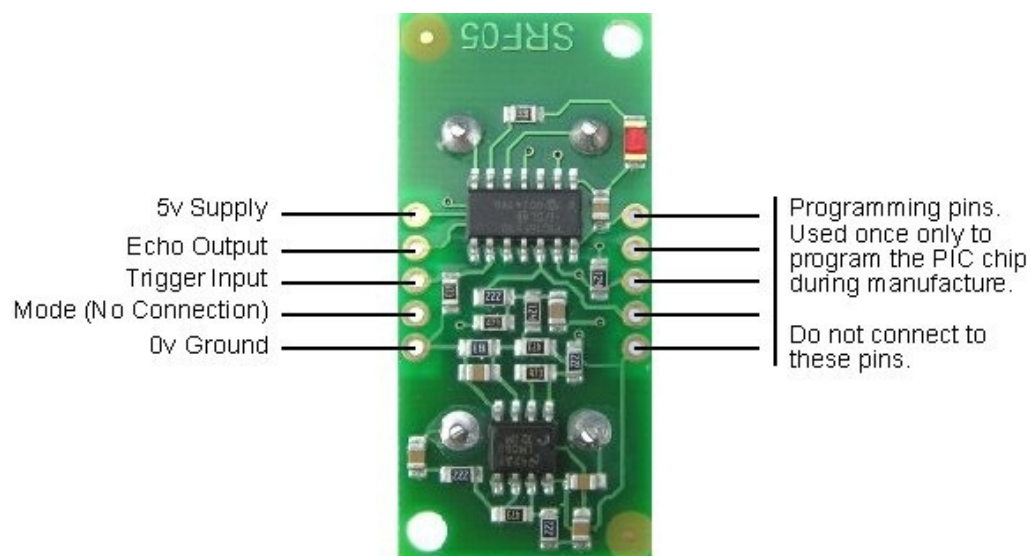
Obr. 35: Řízení serva [13]

Pro generování pulsu pro servomotor je použit Timer1 na mikroprocesoru, kterému se vždy nastaví požadovaná počáteční hodnota a po přetečení vyvolá přerušení. Zdroj hodinového signálu je odvozen od vnitřního taktu procesoru. Timer1 aktivuje přerušení každých cca 13 ms. Dvakrát se vždy nuluje výstup na servomotor a příznakový bit, takže doba mezi pulsy je vždy alespoň 26 ms. V třetím po sobě jdoucím přerušení se výstup na servomotor přepne na log. 1 a nastaví se velikost natočení. Počáteční hodnota v registrech TMR1L a TMR1H je přímo úměrná velikosti natočení servomotoru. Velikost těchto hodnot byla odhadnuta výpočtem a následně experimentálně upravena.



Obr. 36: Blokové schéma Timer 1

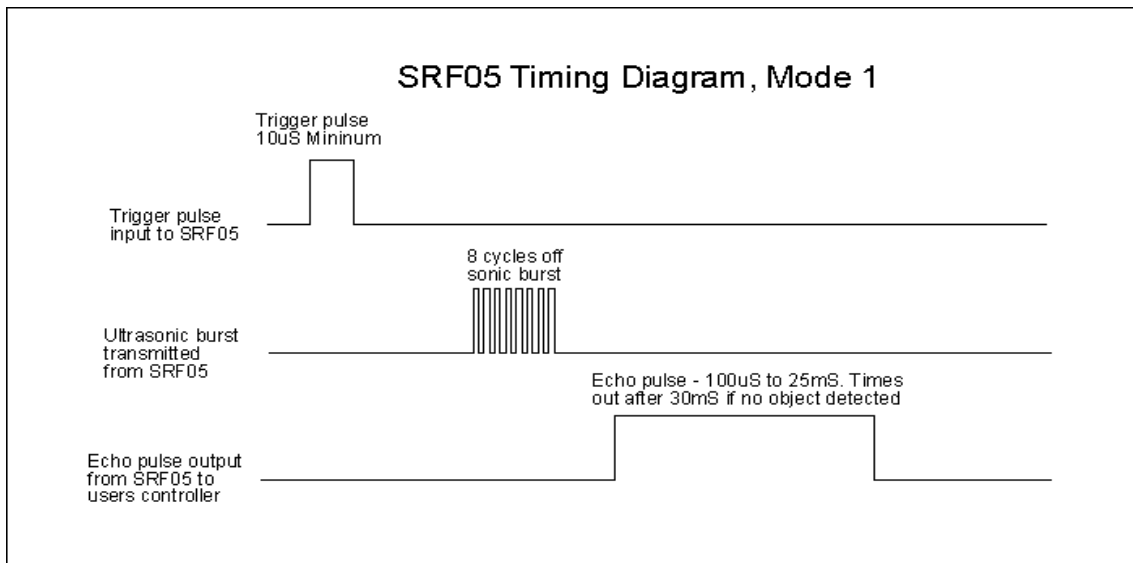
Ultrazvukový sonar SRF05 se obsluhuje pouze pomocí dvou signálů. Další vývody jsou určeny pro napájení nebo nejsou použity viz. obr.37 .



Obr. 37: SRF05 připojení [14]

Signál “Trigger Input“ slouží k aktivaci měření. Tu provedeme tak, že na tento vstup přivedeme impuls o minimální délce 10 us. Po detekci toho impulsu začne sonar vysílat definovaný ultrazvukový signál. Čas mezi odesláním a přijetím ultrazvukového signálu lze zjistit na druhém aktivním pinu “Echo Output“, kde se po dokončení měření objeví

puls o délce úměrné vzdálenosti objektu. Puls delší jak 25 ms znamená, že se před sonarem nenachází žádný objekt. Detaily časování jsou na obr.38.



Obr. 38: SRF05 - časování signálů [14]

Délka echo signálu je detekována programově. Během log. 1 na echo output je aktivní smyčka v níž se inkrementuje proměnná. Tato smyčka trvá 8 programových instrukcí, doba výkonu jedné instrukce je 0,2 us. Jeden průběh smyčkou tedy trvá 1,6 us. Vzdálenost v centimetrech se pak vypočte podle vztahu níže. Detaily lze vyhledat v [14].

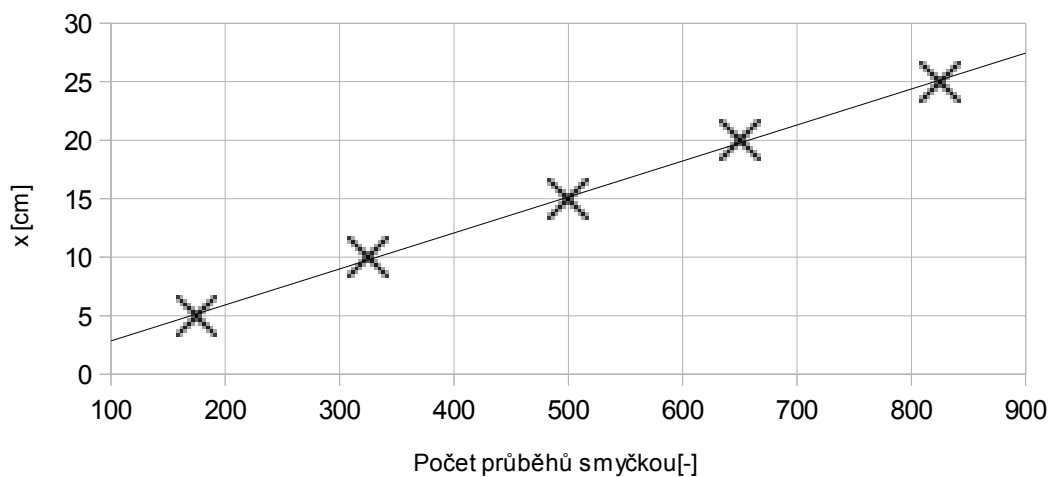
$$x[cm] = \frac{t[us]}{58}$$

Bohužel náš procesor je pouze 8-bitový a počet trvání 1,6 us smyček je většinou větší jak 255, musí se tedy používat dvou proměnných pro uložení výsledku. To má za následek různou délku průběhu smyčkou, díky testování zda proměnná s menším bitovým významem přetekla.

Proto jsme se rozhodli provést ruční kalibraci a pro získání hodnoty v cm následně použít aproximovaný vztah. Několikrát jsme položili předmět před sonar o známe vzdálenosti a odečetli hodnotu počtu průběhů smyčkou. Výsledky jsme zobrazili do grafu, viz. obr.39 a nechali proložit. Rovnici výsledné přímky používáme k výpočtu vzdálenosti v programu na PC.

Délka echa v závislosti na vzdálenosti objektu

sonar SRF05

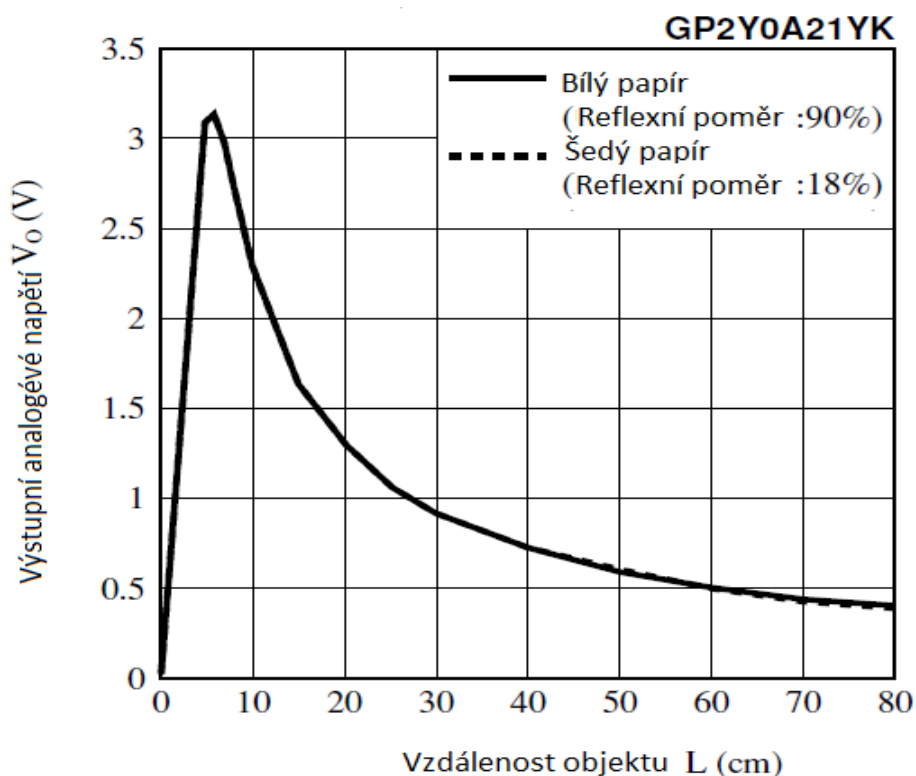


$$f(x) = 0,03x - 0,22$$

Obr. 39: Závislost počtu průběhů smyčkou na vzdálenosti objektu

5.3.5 Senzory GP2Y0A21

Jak už bylo zmíněno dříve, tento typ senzoru je použit nad každým kolem robotu a je určen hlavně ochraně před kolizí s předmětem. Každý senzor je připojen přímo na analogový vstup mikroprocesoru. PIC16F877A umožňuje snímat až osm A/D kanálů s deseti bitovou přesností. Pro získání naměřených hodnot jsme použili knihovnu ADC. Nejprve se musí příslušný pin nakonfigurovat jako analogový a následně stačí zavolat funkci `ADC_Read(káanal)`, která vrátí hodnotu A/D převodu na zvoleném kanálu. Jelikož je výsledek deseti bitový, tak před odesláním do PC se dělí čtyřmi a tak nám stačí na každý senzor pouze jeden byte.



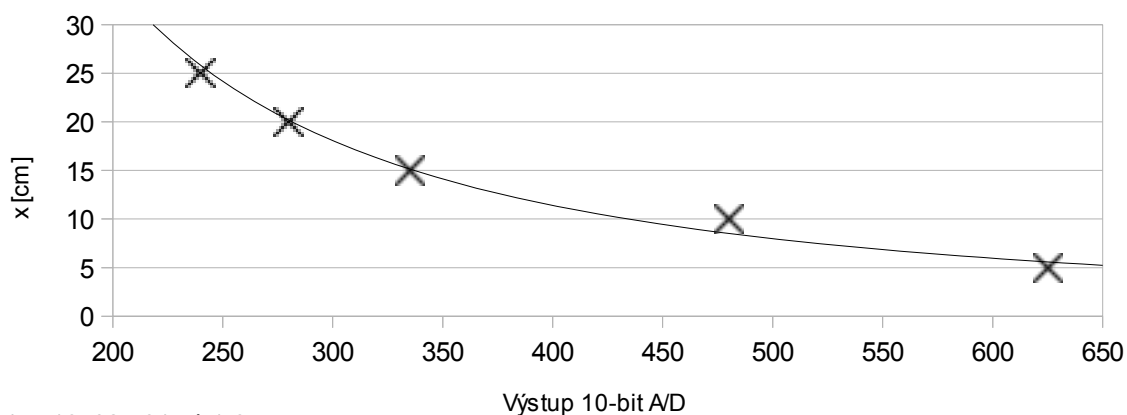
Obr

. 40: GP2Y0A21 - závislost výstupního napětí na vzdálenosti

Výstup z GP2Y0A21YK je nelineární, jak vidět z grafu na obr. 40. Efektivní rozsah použití je přibližně 10 až 80 cm. Pro zobrazení vzdálenosti v cm na PC jsme opět provedli kalibraci, výsledek zobrazili do grafu a nechali proložit. Výsledek je vidět na obr. 41. Rovnice proložení byla také implementována do ovládacího programu v PC.

Závislost výstupního napětí na vzdálenosti objektu

senzor GP2Y0A21

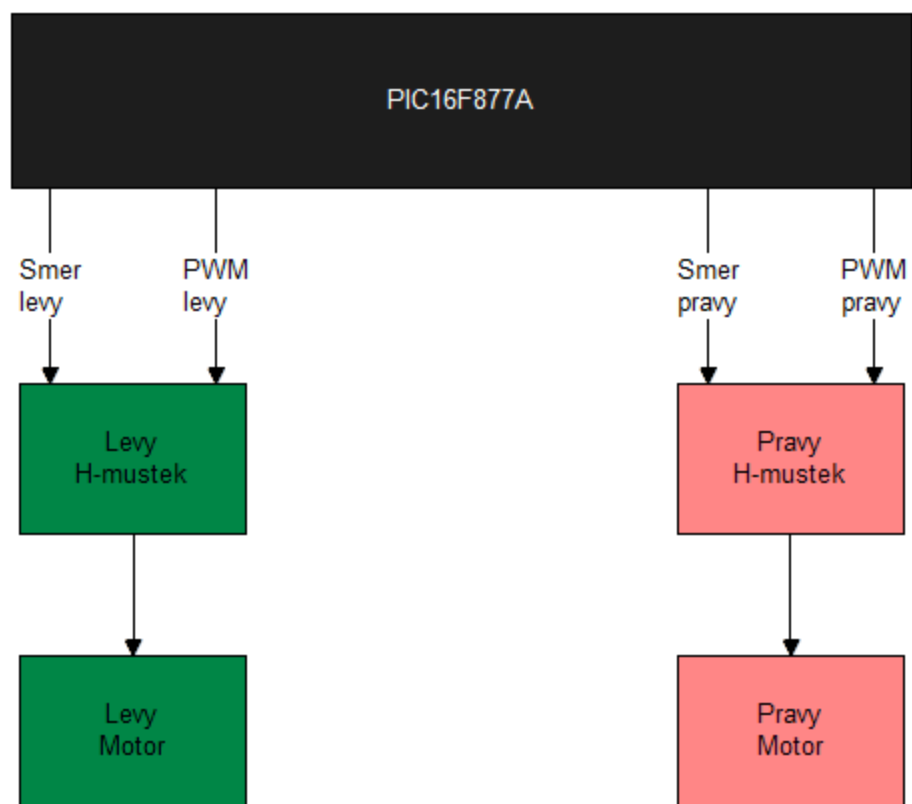


$$f(x) = 164025,81 x^{-1,6}$$

Obr. 41: Graf závislosti vzdálenosti na hodnotě z výstupu A/D převodníku

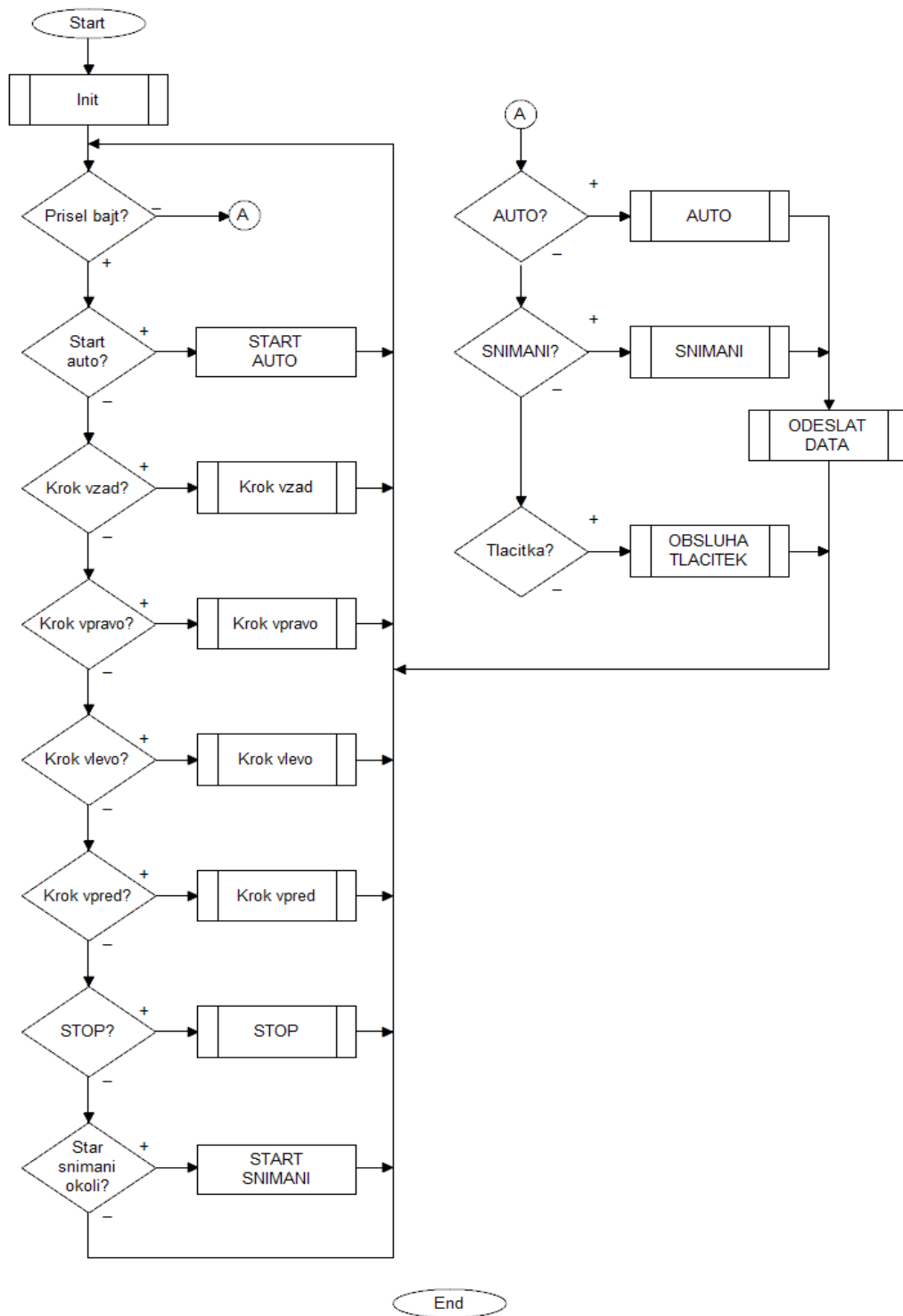
5.3.6 Řízení H-můstek

Pro řízení motorů jsou na robotu umístěny dva H-můstky. Způsob jejich ovládaní je naznačen na obr. 42. Mikroprocesor PIC16F877A obsahuje dva pulzně šířkově modulované kanály. Každý z nich připojen na jeden H-můstek. Střída jednoznačně určuje rychlost otáčení motoru v rozsahu 0 až 100%. Směr otáčení je určen logickou úrovní na druhém vstupu H-můstku.

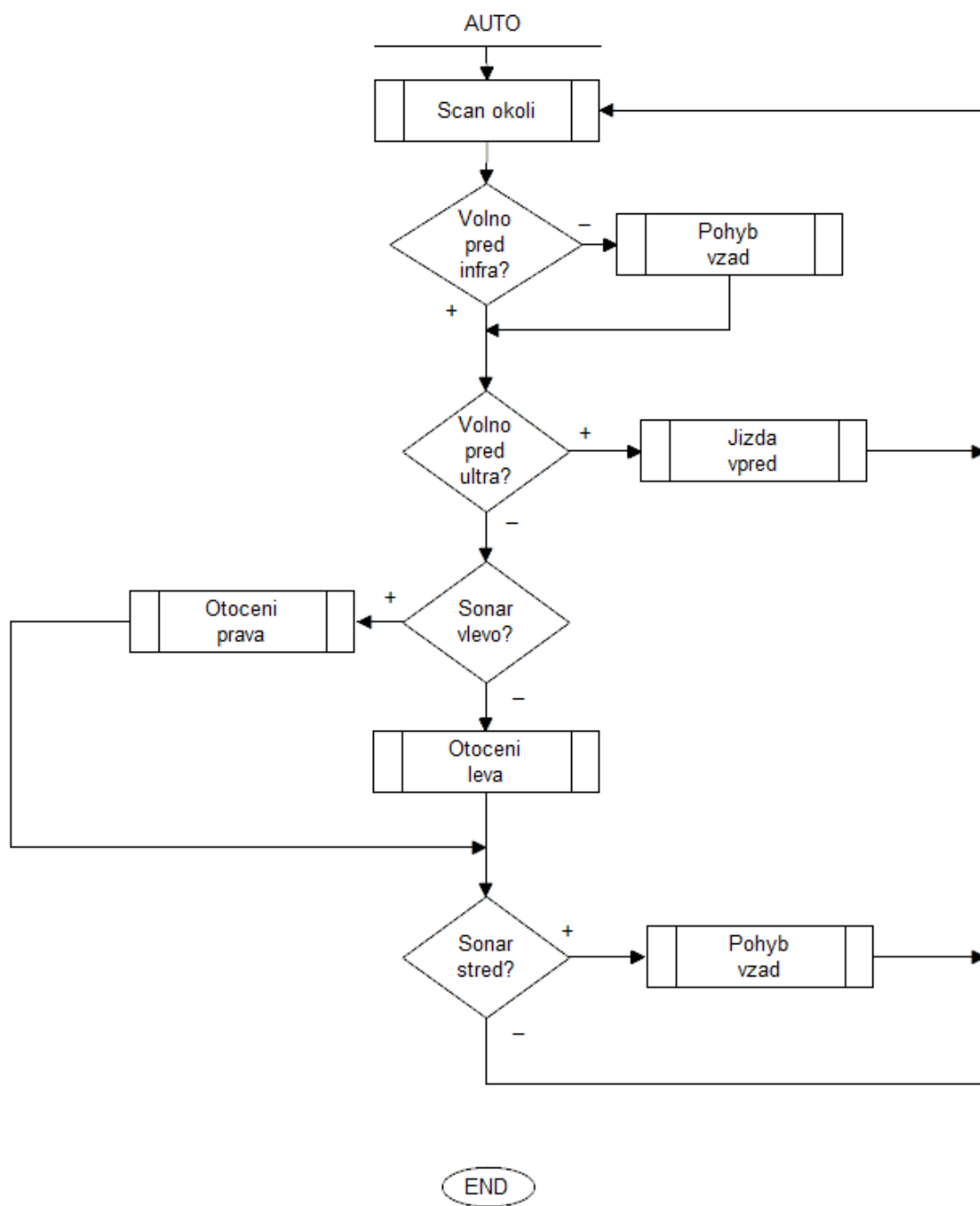


Obr. 42: Připojení H-můstků

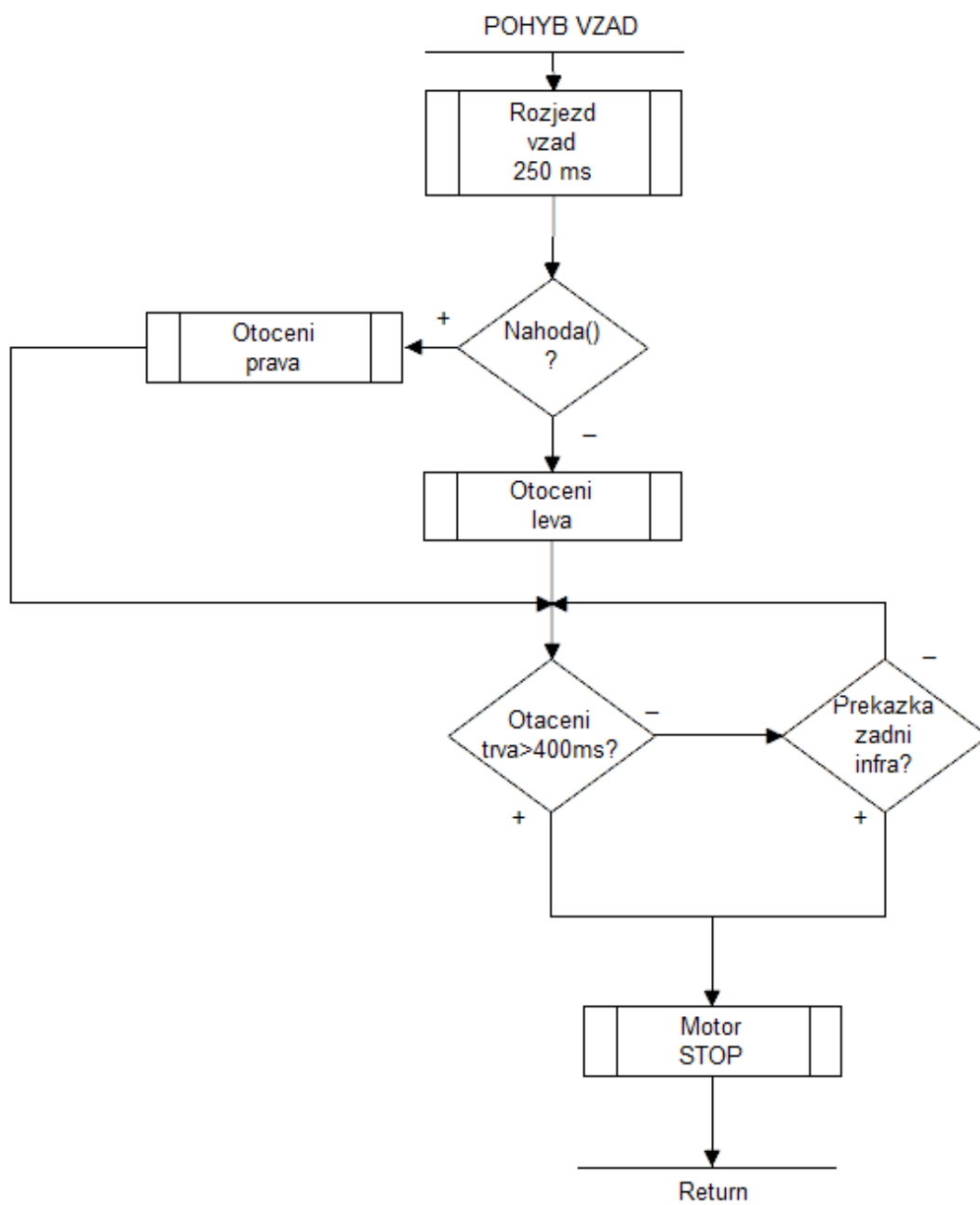
5.3.7 Vývojové diagramy



Obr. 43: Diagram hlavní smyčky



Obr. 44: Diagram automatického režimu



Obr. 45: Diagram pohybu vzad

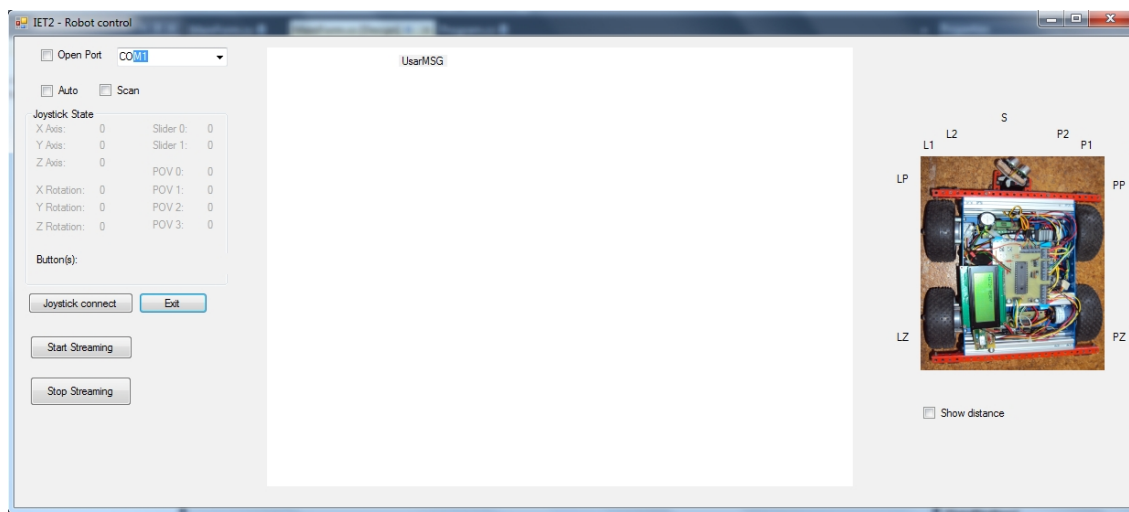
5.4 Program PC

Ovládací program do PC byl vytvořen ve školní licenci vývojového prostředí Visual C# 2010. Pro vývoj aplikace byly použity standardní knihovny, pouze k obsluze joysticku jsme využili SlimDX knihoven.

Pomocí vytvořeného programu se lze připojit k robotu, sledovat hodnoty na senzorech, řídit ho joystickem a zadávat jednoduché povely. Na robotu je též umístěna kamera, která umožňuje sledovat prostředí před robotem. Problematikou týkající se přenosu videa se zabýval kolega Humaj.

5.4.1 Popis programu

V levé horní části si můžeme vybrat sériový port pro komunikaci a následně otevřít spojení. Pokud jsme spojeni s robotem, můžeme zkusit zapnout snímání okolí nebo autonomní režim. Pro ruční řízení lze připojit joystick pomocí panelu níže. Pod tímto panelem nalezneme tlačítka pro start/stop stream videa z kamery. Přenášené video se zobrazuje uprostřed celého programu. A v pravé části programu jsou indikovány vzdálenosti objektů od jednotlivých senzorů v centimetrech.



Obr. 46: Ovládací program v PC

6 ZÁVĚR

V této bakalářské práci se nám podařilo vytvořit přehled o robotických podvozcích a na základě jejich vlastností vybrat vhodný podvozek pro náš projekt. Dále jsme se zabývali různými senzory a zjišťovali co s nimi lze naměřit. Návrh sensorického systému už potom nebyl takový problém. Stačilo si stanovit jaké objekty potřebujeme detekovat a ze znalosti vlastností čidel jsme zvolili potřebnou koncepci.

Vybrali jsme vhodný mikroprocesor pro řízení všech periférií na robotovi. Pro zvolený PIC16F877A jsme navrhli obvodové řešení a jeho funkčnost ověřili na nepájivém poli. Dále jsme navrhli desku plošných spojů, nechali ji vyrobit a provedli celkové oživení funkčnosti. Implementovali jsme softwarové řešení do mikroprocesoru tak, aby vyhovovalo našim potřebám. Robot je schopen komunikace s operátorem, reagovat na jednotlivé povely a předávat informace o prostředí. Námi vytvořený firmware obsahuje i jednoduché automatické demo. Robot se v tomto režimu dokáže samostatně vyhýbat překážkám.

Nakonec byl vytvořen ovládací program do PC pro možnost vzdálené kontroly nad robotem. Pomocí programu lze robota jednoduše řídit, zadávat mu povely a sledovat vlastnosti prostředí.

Celkově se nám podařilo prakticky realizovat celého funkčního robota, který je schopen pohybu v průmyslovém prostředí. Zvolená koncepce umožňuje snadné rozšíření o další periferie jako je například plánovaná robotická ruka. Intuitivní operátorské řízení je zajištěno kombinací joysticku a webkamery.

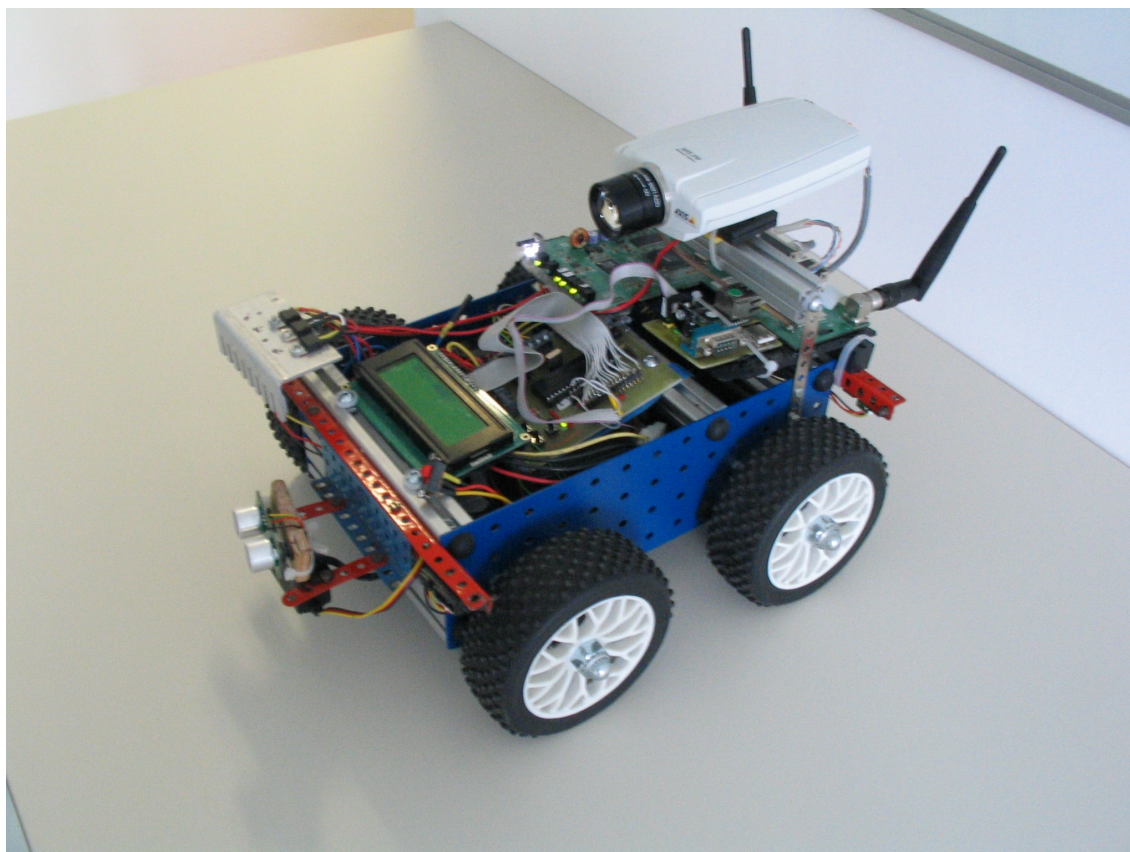
Cíle zadání se podařilo splnit.

6.1 Návrhy na vylepšení

1. **Výkonnější pohonná jednotka** – použitý komerční podvozek byl dodán včetně motorů. Po montáži všech komponent robot uveze sotva sám sebe a možnosti, které nám umožňuje pulzně šířková modulace zůstávají nevyužity. Autonomní režim by mohl být potom určitě zajímavější.
2. **Použití interních senzorů** – doplnění robota o enkodéry natočení, kompas, akcelerometr nebo jejich kombinaci by umožňovalo jak plánování, tak pamatování ujeté trasy. V průmyslu by tato koncepce jistě našla široké uplatnění.

3. **Lepší mikroprocesor** – namísto univerzálního PIC16F877A, použít například dsPIC30F3011. Je vhodný pro řízení až šesti motorů, obsahuje moduly pro připojení kvadratických enkodérů, takže čtením jednoho registru dostáváme rovnou hodnotu natočení. Umožňuje také prioritní přerušení.

6.2 Foto hotového robota



Obr. 47: Hotový robot

Literatura

- [1] Novák, P. Mobilní roboty - pohony, senzory, řízení. Praha : BEN – technická literatura, 2005. ISBN 80-7300-141-1
- [2] Mikroprocesorová technika [online]. 2009 [cit. 2010-05-20]. BMPT. Dostupné z WWW:<<http://www.urel.feec.vutbr.cz/BMPT/index.php?strana=5&lang=CS>>.
- [3] *Snail Instruments* [online]. 2011 [cit. 2011-12-10]. Dostupné z WWW: <<http://shop.snailinstruments.com/>>.
- [4] *MERKUR TOYS s.r.o.* [online]. 2011 [cit. 2011-12-10]. Dostupné z WWW: <<http://www.merkurtoys.cz/>>.
- [5] *Cognitoware : Robotics Software Libraries* [online]. 2009 [cit. 2011-12-10]. Bayes Filters for a Differential Drive Robot. Dostupné z WWW: <<http://cognitoware.com/tutorials/DifferentialDrive.htm>>.
- [6] *Robotthoughts* [online]. 2011 [cit. 2011-12-10]. Astro Droid Synchro Drive. Dostupné z WWW: <<http://www.robotthoughts.com/lego/2000/07/astro-droid-synchro-drive/>>.
- [7] Sensor Design [online]. 2011 [cit. 2011-12-10]. Sensor Design. Dostupné z WWW:<<http://www.clear.rice.edu/elec201/Book/sensors.html>>.
- [8] Wikipedia [online]. 2011 [cit. 2011-12-10]. Ackermann steering geometry. Dostupné z WWW: <http://en.wikipedia.org/wiki/Ackermann_steering_geometry>.
- [9] ĎAĎO, S; KREIDL, M. *Senzory a měřicí obvody*. Praha : Vydavatelství ČVUT, 1999. ISBN 80-01-02057-6
- [10] BEJČEK, L. *Měření neelektrických veličin : Skriptum*. Brno : VUT Brno, 1988. 241 s.
- [11] *PIC16F87XA : Data Sheet [online]. [s.l.] : Microchip Technology Inc., 2003 [cit. 2011-12-11]. Dostupné z WWW: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>>.*
- [12] *BUCKLEY, David. Robotsystems.net [online]. 2009 [cit. 2011-12-12]. Simple Walker Mechanisms. Dostupné z WWW: <<http://www.davidbuckley.net/RS/WalkerMechanisms/SimpleWalkerMechanisms.htm>>.*

- [13] *Pojezdy.eu: Modelářská serva - základní informace. Pojezdy.eu [online]. 21. 07.2011 [cit. 2012-04-16]. Dostupné z WWW:*
<<http://www.pojezdy.eu/view.php?cisloclanku=2011070004>>
- [14] *ROBOT Electronics: SRF05 - Ultra-Sonic Ranger Technical Specification. ROBOT Electronics [online]. [cit. 2012-04-16]. Dostupné z WWW:*
<<http://www.robot-electronics.co.uk/htm/srf05tech.htm>>
- [15] *mikroElektronika: Development tools, Compilers, Books mikroElektronika [online]. [cit. 2012-04-18]. Dostupné z WWW: <<http://www.mikroe.com/>>*

SEZNAM PŘÍLOH

- Příloha 1 Výpis programu do PIC16F877A
Příloha 2 Obsah CD disku

Příloha 1

```
/*
 * Project name:
 *   IET2 Control Board
 * Copyright:
 *   (c) Jakub Hulka, 2011-2012
 * Application configuration:
 *   MCU:          PIC16F877A
 *   Oscillator:   HS, 20.0000 MHz
 *   Ext. Modules: -
 *   SW:           mikroC v2.5
 */

#define UP PORTD.F7           //button UP
#define DOWN PORTD.F6        //button DOWN
#define ENTER PORTD.F5       //button ENTER
#define LED1 PORTD.F3        //LED1-green
#define LED2 PORTD.F4        //LED2-red
#define Ser_OUT PORTD.F0      //Output for servo
#define Echo_output PORTD.F2 //Input from sonar echo
#define Triger_input PORTD.F1 //Output for sonar triger
#define AngleStep 7          //Size of serve step
#define LeftPWM PORTC.F1     //Left motor speed
#define LeftCourse PORTC.F0  //Left motor diretion
#define RightPWM PORTC.F2    //Right motor speed
#define RightCourse PORTC.F3 //Right motor diretion
#define Speed 255           //Init motor speed
#define MaxSpeed 255
#define STOP 350

unsigned test_angle=17;
unsigned edge=0;
int PP, LP, PZ, LZ, Ultrasonic[5];
unsigned pom;
unsigned run=0, scan=0;
unsigned leftS=0, rightS=0;
unsigned smer=0;
int temp;

// Lcd pinout settings
sbit LCD_RS at RB1_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D4 at RB4_bit;

// Pin direction
sbit LCD_RS_Direction at TRISB1_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D7_Direction at TRISB7_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB4_bit;
```

```

unsigned UltraTest()
{
    unsigned cnt=0;
    Triger_input=1;    //Start pulse
    Delay_10us();
    Triger_input=0;    //Start pulse
    while(!Echo_output); //Wait for high
    while(Echo_output) //Count high time
    {
        cnt++;
    }
    return cnt;    //Return high time
}

void interrupt()
{
    edge++;    //Update servo level
    switch(edge)
    {
        case 1:    //Servo level low 13 ms
        {
            Ser_OUT=0;
            TMR1L=0x00;
            TMR1H=0x00;
            break;
        }
        case 2:    //Servo level high
        {
            Ser_OUT=1;    //Set high time to
            TMR1L=0x00;
            TMR1H=0xD0+test_angle; //To define angle
            break;
        }
        case 3:    //Servo level low 13 ms
        {
            Ser_OUT=0;
            TMR1L=0x00;
            TMR1H=0x00;
            edge=0;
            break;
        }
        default:
            edge=0;
            break;
    }
    PIR1.TMR1IF = 0;    //Cler interupt flag
}

void ScanGP(void)
{
    //Read all A/D channel
    LZ = Adc_Read(4);
    PP = Adc_Read(3);
    LP = Adc_Read(2);
    PZ = Adc_Read(1);
}

```

```

void Back(void)
{
    LeftCourse=1;
    RightCourse=1;
    PWM2_Set_Duty(MaxSpeed);
    PWM1_Set_Duty(MaxSpeed);
    Delay_ms(250);
    if(rand()>16154)
        RightCourse=0;
    else
        LeftCourse=0;
    for(pom=0;pom<4;pom++)
    {
        Delay_ms(100);
        ScanGP();
        if(PZ<STOP||LZ<STOP)
            continue;
    }
}

void StartAuto(void)
{
    run=1;
    INTCON=0b11000000;
    PWM1_Set_Duty(Speed);
    PWM2_Set_Duty(Speed);
    PWM1_Start();
    PWM2_Start();
    Lcd_Cmd(_LCD_CLEAR); // Clear display
    Lcd_Out_CP("-IET2- AUTO");
}

void StopAuto(void)
{
    run=0;
    INTCON=0b00000000;
    LED1=1;
    LED2=0;
    RightCourse=~RightCourse;
    LeftCourse=~LeftCourse;
    Delay_ms(50);
    PWM1_Stop();
    PWM2_Stop();
    PWM1_Set_Duty(0);
    PWM2_Set_Duty(0);
    Lcd_Cmd(_LCD_CLEAR); // Clear display
    Lcd_Out_CP("-IET2- READY");
    scan=0;
}

```

```

void Scanig(void)
{
    if(test_angle>30)
        smer=1;
    if(test_angle<5)
        smer=0;
    if(!smer)
        test_angle+=AngleStep;
    else
        test_angle-=AngleStep;
    ScanGP();
    Delay_ms(75);
    temp=UltraTest();
    temp=temp/25;

    switch(test_angle)
    {
        case 3:
        {
            Ultrasonic[0]=temp;
            break;
        }
        case 10:
        {
            Ultrasonic[1]=temp;
            break;
        }
        case 17:
        {
            if(smer)
                Ultrasonic[2]=temp;
            break;
        }
        case 24:
        {
            Ultrasonic[3]=temp;
            break;
        }
        case 31:
        {
            Ultrasonic[4]=temp;
            break;
        }
    }
}

void SendData(void)
{
    UART1_Write_Text("#");
    UART1_Write(PP/4);
    UART1_Write(LP/4);
    UART1_Write(PZ/4);
    UART1_Write(LZ/4);
    UART1_Write(Ultrasonic[0]);
    UART1_Write(Ultrasonic[1]);
    UART1_Write(Ultrasonic[2]);
    UART1_Write(Ultrasonic[3]);
    UART1_Write(Ultrasonic[4]);
}

```

```

void MotorStar(void)
{
    PWM2_Set_Duty(Speed);
    PWM2_Start();
    PWM1_Set_Duty(Speed);
    PWM1_Start();
}

void RotSon(void)
{
    INTCON=0b11000000;
    Delay_ms(250);
    INTCON=0;
}

void main()
{
    //-----INIT-----
    char text[] = "-IET2- READY";
    srand(43);
    PORTA = 0;           //RA4,5 not use
    TRISA = 0xFF;       //RA0-RA4 input for GPY021A
    PORTB = 0;
    TRISB = 0x01;      //RB0-not use,RB1-RB7 display
    PORTC = 0;
    TRISC = 0x00;      //RC0-RC4 2xH-mustek, RC5-RC7 not use
    PORTD = 0;         //RD0-RD2 not use
    TRISD = 0xE4;      //RD3-led1,RD4-led2,RD5-RD7 buttons
    PORTE = 0;
    PORTE = 0x0F;      //not use
    ADCON0=0x01;       //Config A/D
    ADCON1= 0x40;      //Channels
    PIE1=0x01;         //Enable periferial interupt
    T1CON=0b11000001; //Config timer1

    Lcd_Init();
    Lcd_Cmd(_LCD_CLEAR); // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
    Lcd_Out(1,1,text);
    LED1=1;
    LED2=0;
    PWM1_Init(1500);
    PWM2_Init(1500);
    PWM1_Set_Duty(0);
    PWM2_Set_Duty(0);
    RightCourse=0;
    LeftCourse=0;
    UART1_Init(57600);
    //-----END OF INIT-----

    //-----MAIN LOOP-----
    while(1)
    {
        //-----DATA RECEIVED-----
        if(UART1_Data_Ready())
        {
            //Read received byte
            pom=UART1_Read();
        }
    }
}

```



```

switch(pom)
{
  case 0x30:{
    StartAuto();
    break;
  }
  case 0x31:{
    StopAuto();
    break;
  }
  case 'A':{
    LeftCourse=0;
    RightCourse=0;
    MotorStar();
    break;
  }
  case 'B':{
    LeftCourse=1;
    RightCourse=1;
    MotorStar();
    break;
  }
  case 'C':{
    LeftCourse=0;
    RightCourse=1;
    MotorStar();
    break;
  }
  case 'D':{
    LeftCourse=1;
    RightCourse=0;
    MotorStar();
    break;
  }
  case 'E':{
    if(test_angle<30)
    test_angle+=AngleStep;
    RotSon();
    break;
  }
  case 'F':{
    if(test_angle>5)
    test_angle-=AngleStep;
    RotSon();
    break;
  }
  case 'G':{
    scan=1;
    INTCON=0b11000000;
  }
}
}
//-----END OF DATA RECEIVED-----

```

```

if(!ENTER)
{
    StartAuto();
}
if(!UP)
{
    StopAuto();
    scan=0;
}
if(!DOWN)
{
    scan=1;
    INTCON=0b11000000;
}
if(scan)
{
    Scanig();
    Delay_ms(50);
}
if(run)
{
    Scanig();
    Delay_ms(50);
    if(PP>STOP||LP>STOP)
    {
        Back();
    }
    if(temp<50)
    {
        if(test_angle<20)
        {
            LeftCourse=1;
            MotorStar();
        }
        else
        {
            RightCourse=1;
            MotorStar();
        }
        if(test_angle==17)
        {
            Back();
        }
        Delay_ms(150);
    }
    else
    {
        RightCourse=0;
        LeftCourse=0;
        MotorStar();
    }
}
if(scan||run)
    SendData();
    Delay_ms(50);
}
//-----END OF MAIN LOOP-----
}

```

Příloha 2

./Bakalářská práce/

./DPS/

./Program PIC/

./Program PC/

Elektronická verze bakalářské práce

Podklady pro výrobu DPS (Eagle)

Zdrojový kód pro PIC16F877A

mikroC for PIC

Zdrojové kódy Ovládacího programu v PC