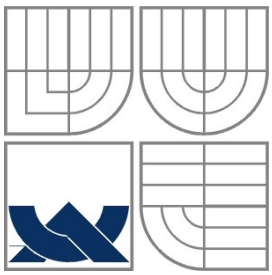


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

VYTVOŘENÍ ŘÍDICÍHO SOFTWARE PRO AUTONOMNÍ MOBILNÍ ROBOT

CONTROL SOFTWARE FOR AUTONOMOUS MOBILE ROBOT

BAKALÁŘSKÁ PRÁCE
BACHELOR THESIS

AUTOR PRÁCE **JAKUB VOJTA**
AUTHOR

VEDOUCÍ PRÁCE **ING. STANISLAV VĚCHET, PH.D.**
SUPERVISOR

BRNO 2010

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

(na místo tohoto listu vložte originál a nebo kopii zadání Vaší práce)

LICENČNÍ SMLOUVA

(na místo tohoto listu vložte vyplněný a podepsaný list formuláře licenčního ujednání)

ABSTRAKT

Bakalářská práce je zaměřena na tvorbu software pro mobilní autonomní robot s mikrokontolérovou řídící jednotkou. Poskytuje přehled senzorů pro mobilní pozemní roboty na úrovni nutné pro programátora řídícího software. Dále rozebírá mikrokontroléry, jejich architekturu a programování. Dále je teoreticky popsán nasazený navigační software a protokol komunikace řídící jednotky robotu s počítačem.

ABSTRACT

Bachelor thesis focuses on software development for mobile autonomous robot with a microcontroller control unit. It provides an overview to sensors for mobile ground robots on a level necessary for control software programmer. It further analyzes microcontrollers and their architecture and programming. At the end of the thesis is theoretical description of deployed navigational software and communication protocol between control unit and computer.

KLÍČOVÁ SLOVA

mobilní, autonomní, robot, mikrokontrolér, mikropočítač, programování

KEYWORDS

mobile, autonomous, robot, microcontroller, microcomputer, programming

PODĚKOVÁNÍ

Děkuji Ing. Stanislavu Věchetovi, Ph. D. za odborné vedení, cenné připomínky a rady.

Obsah:

1 ÚVOD

Schopnost vnímat okolí a měnit podle toho své chování patří k hlavním rysům autonomních mobilních robotů. Bez vjemů zprostředkovaných senzory je robot jen automatem procházejícím naprogramovaný úkol stále dokola a bez obměny. Taková zařízení jsou jistě užitečná pro dnešní průmysl, ale roboty vybavené senzory a inteligentním software nabízejí o mnoho více. Mohou se přizpůsobovat prostředí nebo pohybovat v nebezpečných prostorách. Mohou dokonce reagovat na člověka a jednoduchým způsobem s ním komunikovat, a to je vlastnost velmi přitažlivá nejen pro režiséry vědeckofantastických filmů.

Lidské bytosti vnímají svět kolem sebe hlavně zrakem a často si nedokáží představit typy smyslů, s jakým si musí vystačit roboty. Hmyz, jako mouchy a pavouci mají také složené oči s mnoha tisíci segmenty reagujícími na světlo. Ale mají například i jemné chloupky, které vnímají změny v proudění vzduchu se zpožděním několika milisekund. Proto je tak těžké je chytit.

Dnešní robotika jako obor zabývající se navrhováním, konstrukcí a přiváděním robotů k životu je polem velmi širokého záběru. Aby robot nezůstal pouhým plánem na papíře, drahou hračkou, nebo nefunkční změtí materiálu, musí být splněna poměrně mocná množina předpokladů. Kromě nezbytných technických znalostí, které mohou být rozprostřeny v týmovém prostředí, jde třeba i o schopnost takový tým řídit, trpělivost, umění improvizovat, kontakty s lidmi, kteří produkt potřebují a na neposledním místě dostatek času a finančních prostředků. Nakonec bohužel ani zdánlivý dostatek všeho potřebného nemusí stačit a celá snaha vyjde jako nevydařený podnik. Pokud jde o mobilní autonomní roboty, knihy jich popisují desítky, ale jen zlomek jich našel skutečné uplatnění. Obvykle bylo navíc potřeba vyšších společenských cílů, například nutnost zacházet s výbušninami, aby bylo dosaženo uspokojivých výsledků. I dnes, tedy po desetiletích výzkumů a vývoje, je aplikace mobilních autonomních robotů napůl snem.

Naději na širší nasazení mobilních autonomních robotů dává rozvoj počítačových technologií posledních let, jejich rostoucí výkon a s tím související nové možnosti software. Díky uvádění novinek v technologiích rovněž klesá cena integrovaných počítačů – mikrokontrolérů. Zejména u nich spadly finanční a znalostní hráze natolik, že vzniklo tisíce malých projektů založených na této minimalistické výpočetní elektronice. Řídící a výpočetní elektronikou se zabývá kapitola 2.2 této práce.

Aby se programovatelná elektronika mohla zdát inteligentní, musí mít promyšlené programy. Aby bylo možné se zabývat programy, musel dříve jejich návrhář mít hluboké znalosti programování v jazycích takzvané nízké úrovně. To znamená zvládnout programování blízké strojovým instrukcím, tedy takřka přímým pokynům hardware počítače. To vyžadovalo vysokou soustředěnost a inteligenci. Dnes jsou nad tímto nadřazené vyšší strukturované jazyky, není nutné detailně rozumět hardware počítače, abychom mohli začít programovat. Tvorba software je dostupnější, přehlednější, rychlejší. Tvorbou software pro mikrokontroléry se věnuje kapitola 2.3.

Automobilový průmysl zase pomáhá vývoji senzorů pro mobilní roboty, jelikož našel uplatnění senzorů při zvyšování bezpečnosti provozu. Senzory používanými pro mobilní roboty se zabývá kapitola 2.1.

2 ROZBOR PROBLÉMU

Úkolem je navrhnout řídící software pro robot pohybující se v přesně specifikovaném prostředí. Musíme proto zvážit vlastnosti senzorů, pracovní prostředí robotu, možnost vzájemného ovlivňování senzorů a náročnost předpokládaných matematických výpočtů. Robot bude vybaven elektromotorovou pohonnou jednotkou, několika infračervenými senzory vzdálenosti, jednočipovou řídící jednotkou a do budoucna i systémem infračervených navigačních bodů v terénu. Lze tedy předpokládat interference infračervených zářičů a problémy s omezenými schopnostmi snímání nejbližšího okolí.

Prostředí působení robotu budou místnosti s rovnými podlahami, zdmi, rozmístěnými krabicemi, stoly, židlemi, případně i osobami v množství odpovídajícím běžné domácnosti.

Komunikace mezi jednotlivými částmi zařízení počítá s rozhraním typu sběrnice. Datový tok je v některých případech obousměrný. Je nutné počítat s určitou úrovní rušení při přenosu a vyvinout ochranné detekční mechanismy.

2.1 Senzory

Zdroje této práce často kategorizují senzory podle třídy robotu, například použitelné v robotu pozemním, podvodním, dálkově ovládaném, autonomním, atp. Tato podkapitola probírá pouze senzory používané v pozemních robotech. Slouží k diagnostice a stavu částí robotu a snímání okolí pro navigaci.

Lze dále rozlišovat senzory dotykové a bezdotykové. Bezdotykový je například radiolokátor využívaný v předpovědi počasí, který obsáhne celou ČR. Dotykový je jednoduchý nárazník odhalující sloupek v naplánované cestě. Autonomní robot má však omezené zdroje energie, musíme tak pečlivě zvážit i spotřebu obou jmenovaných principů.

Každý mobilní autonomní robot by měl být schopen se pohybovat v daném prostředí bez vrážení do předmětů. V praxi má ale jen omezené prostředky pro vnímání povahy a celkových rozměrů překážek. Často se tedy neobejde bez doplnění údajů o okolí od lidské obsluhy. V úvahu připadá buď kompletní zmapování místnosti včetně překážek anebo zavedení zakázaných oblastí, kterým se robot bude vyhýbat.

O mnoho tvrdším oříškem zůstává fungování robotu ve venkovním prostředí, a to z mnoha důvodů. Narůstají náklady na konstrukci terénního provedení, snižuje se spolehlivost v drsných atmosférických vlivech, hrozí hromadění se statické elektřiny, zuří venkovní provoz na chodnících a silnicích. Práce robotu mimo budovy vyžaduje mnoho zodpovězených technických otázek, než-li je vůbec možné začít přemýšlet nad inteligencí celého systému.

Pozitivní zprávou pro konstruktéry robotů je neuhasínající rozvoj této oblasti. Oblast senzorů vhodných pro mobilní roboty zažívá rozvoj díky snahám o vyšší bezpečnost v autodopravě. Mluvíme o inteligentních automobilech.

2.1.1 Vybrané vlastnosti senzorů

Dosah – senzory pro detekci překážek mívají maximální a minimální hodnoty dosahu, často v intervalu několika desítek centimetrů. Nelze tak detekovat překážky bližší než minimální a vzdálenější než maximální dosah senzoru.

Typ detekovaných překážek – některé objekty neodrážejí ideálně všechnu vysílanou energii, nebo ji naopak odrážejí několikrát v prostoru. Dochází tak ke zkreslování měřených údajů.

Rušení – měli bychom zajistit, aby se senzory pracující na stejném principu navzájem nerušily. Hlídat bychom měli i rušení dalšími elektromagnetickými obvody, typicky např. rušení kompasu elektromotorem.

Nadbytečnost – pokud při provozu dojde k výpadku některého senzoru, měla by jeho funkce být alespoň částečně nahrazena jinými senzory.

Jednoduchost, cena – hledáme senzory cenově dostupné, jenž se snadno shánějí a vyměňují.

Spotřeba – již ve fázi plánování lze z informací od výrobce zjistit spotřebu el. energie, zpravidla ve Volt-ampérech a porovnat s kapacitou a doporučeným odběrem zdroje.

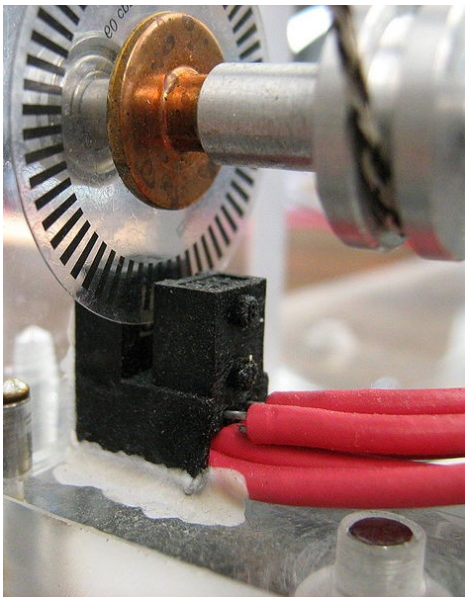
Velikost – výrobce většinou poskytuje zjednodušený výkres senzoru a jeho hmotnost. Tyto musí být praktické vzhledem k zamýšlenému rozměru robotu.

Obtížnost zpracování výstupních dat – zda se jedná o signál napěťový, nebo proudový, jaké jsou jeho úrovně.

2.1.2 Senzory používané v mobilních robotech

Diagnostické – snímají stav periferií robotu, teplotu nebo zbývající napětí na bateriích, množství radiace, sílu radiového signálu s řídícím centrem.

Otáčkoměry, senzory natočení – snímají rychlosti otáčení a okamžité natočení motorů. Na základě těchto údajů je řídící jednotka schopna vypočítat polohu a rychlost robotu. Pro výpočet polohy je tedy nutno znát informaci o hřídeli motoru a zajistit pevný kontakt kol s podlahou. Senzory natočení jsou digitální a analogové. Oba druhy využívají změn elektrických veličin, každý však jiným způsobem. Analogový je založený na měření napětí generovaného otáčejícím se motorem, který funguje jako dynamo. Digitální senzory natočení využívají fotoelektrických, magnetických a třecích jevů. Příklad fotoelektrického snímače natočení lze najít v téměř každé počítačové myši. Jde o pravidelně děrované mezikruží, které slouží jako přerušovač mezi LED a fototranzistorem.

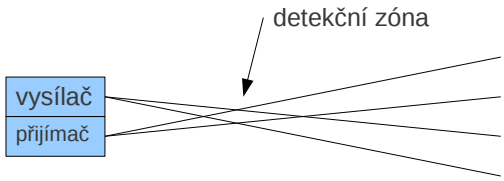


Obr. 1 model inkrementálního snímače, zdroj [3]

Sonary – (z anglického sound navigation and ranging) skládá se z vysílače a přijímače. Vysílají zvukový signál a přijímají jeho odraz. Vypočtená vzdálenost je potom funkcí času uběhlého mezi vysláním a přijetím.

Taktilní senzory – snímají dotyk s překážkou, nejčastěji jde o formu pohyblivě upevněného nárazníku, který se při kontaktu stlačí.

Infračervený senzor vzdálenosti – sestává z LED a fototranzistoru, tedy vysílače a přijímače. Oblíbená konstelace je naklonit optické osy vysílače a přijímače tak, že se protínají v určité vzdálenosti od senzoru. Vznikne detekční zóna, ve které lze určit poměrně přesně překážku a její vzdálenost.

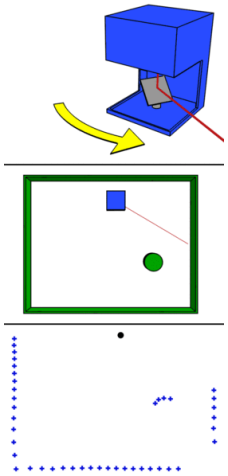


Obr. 2 naklonění optických os u infračerveného senzoru

Kompas – obsahuje několik snímačů změny magnetického pole Země, údaje z nich zpracovává řídící jednotka.

GPS – (Globální poziční systém) využívá soustavy družic (satelitů) na oběžné dráze Země. Družice vysílají údaj o čase, svoji polohu na orbitě a korekční údaje. Každý přijímač GPS je pak na základě času schopen určit vzdálenost od satelitu. Pokud má přijímač údaje z alespoň čtyř satelitů, je schopen vypočítat zeměpisnou šířku, délku a nadmořskou výšku.

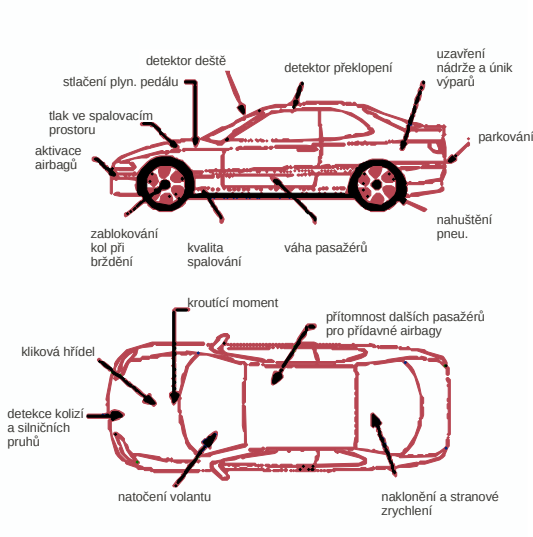
Laserové scannery – naměřená vzdálenost je opět funkcí času, paprsek je vyslán a přijat jeho odraz. Rychlost světla je vysoká, tento způsob není vhodný pro přesnost s odchylkou menší, jak 1mm. Používají se v průmyslu ke kontrole výrobků. Pěknou aplikací laserových senzorů je scanner firmy SICK, kde je doplněno rychle se otáčející zrcadlo. Zrcadlo svírá s osou otáčení úhel 45° a měří vzdálenosti v kruhové výseči.



Obr. 3 funkce laserového 2D scanneru, zdroj [3]

2.1.3 Inteligentní automobily

Hlavním přínosem zavádění inteligentních systémů je zvýšení bezpečnosti silničního provozu. Jde o nalezení takových technických řešení, která by zbytečně neodváděla pozornost řidiče, spíše vylepšovala jeho omezené smysly rozptýlené rychlostí v provozu. Výsledkem je řada detektorů uzpůsobených pro automobily. Sledují se překážky před vozidlem, pruhy na vozovce, rychlosti vozidla a jeho poloha. Používají se zejména laserové měřiče vzdálenosti v kombinaci s výpočetní řídící jednotkou. Výsledkem je v současnosti automobil, který zvukově upozorní, pokud řidič kličkuje nebo se příliš blíží k vozidlu před sebou, ale člověk zatím není zcela vyřazen z ovládání automobilu [2].



Obr. 4 Co vše je možné snímat na automobilu, zdroj: [5]

2.2 Řídící jednotka

Obecně hledáme obvod, který bude plnit okruh úkolů požadovaných pro řízení mobilního robotu. Napřed se zjistí, co všechno bude řídící jednotka obsluhovat. Pokud budou připojeny senzory, musí být na jednotce dostatek vstupních a výstupních periferií potřebného standardu. Některé senzory dodávají spojitý signál, jiné impulsy s různou periodou. Data ze senzorů musejí být odečítána a zpracována v krátkých časových intervalech. Dále je potřeba odeslat příkaz pohonům a dalším akčním členům. Musíme prozkoumat provozní prostředí, pro nepříznivé podmínky se používají průmyslová provedení počítačů. Pokud se bude robot pohybovat v terénu, neměl by v řídící jednotce mít prvky s pohyblivými částmi.

Díky vývoji ve výpočetní technice je na výběr množství procesorů, různou měrou integrovaných a výpočetně silných. Protože navrhovaný robot bude mít poměrně málo senzorů a akčních členů, bude prozkoumána oblast mikrokontrolérů.

2.2.1 Mikrokontrolér

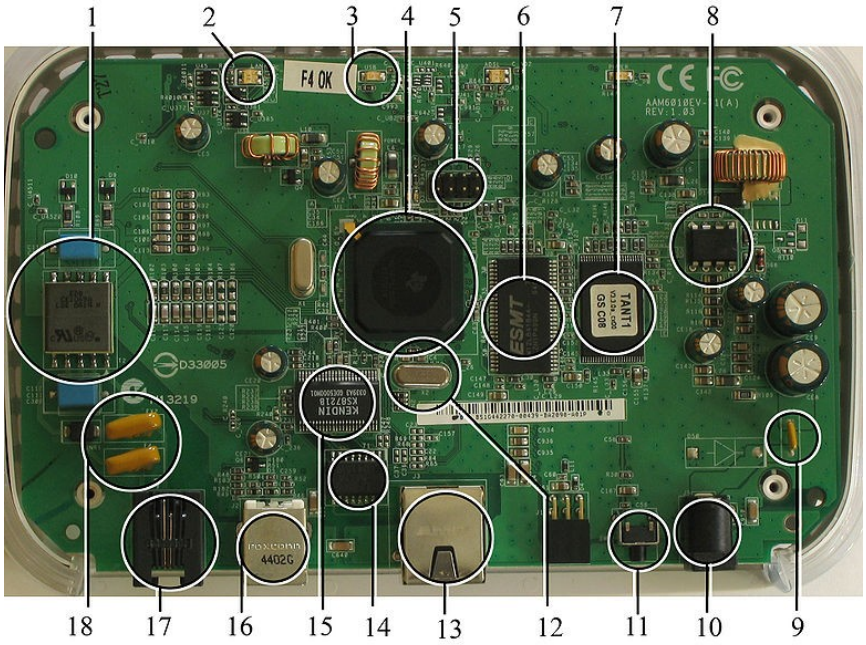
Mikrokontrolér je polovodičová součástka vyráběná nejčastěji jako integrovaný obvod. Můžeme si ji představit jako miniaturní počítač. Základním prvkem tohoto počítače je mikroprocesor, dále pak paměti a vstupně-výstupní bloky. Jak bude počítač pracovat je uloženo v programu v paměti. Při řešení rozdílných úloh je třeba měnit pouze program.

Vznik mikrokontrolérů byl motivován především snahou o plné využití technologických možností výroby integrovaných obvodů. Původně byly místo nich používány logické obvody vyrobené na zakázku. V zakázkových obvodech, navrhovaných vždy pro konkrétní úlohu, se funkčnosti dosahovalo určitým uspořádáním logických členů (hradel) a klopných obvodů. To je značně nákladný způsob, vyplatil se tedy pouze v sériové výrobě. Vznik mikrokontrolérů se datuje do počátku 70. let.

2.2.2 Oblasti aplikace mikrokontroléru

Mikrokontroléry jsou téměř v každé dnešní spotřební elektronice jako tzv. vestavěné systémy. Na obrázku je příklad vestavěného systému ADSL modem a router. Mezi vyznačenými součástmi je procesor (4), RAM (6) a flash paměť (7).

Vestavěným systémem můžou být náramkové hodinky hudební přehrávače, ale i třeba automatické pračky. Na rozdíl od všestranných osobních počítačů, design vestavěného systému je podřízen plnění úzké množiny úkolů.



Obr. 5 Vestavěný systém, zdroj: [3]

2.2.3 Architektura mikrokontroléru

Mikrokontroléry se jako složitá zařízení skládají ze tří hlavních bloků:
Operační paměti – ukládají se zde data a program
Vstupně-výstupní blok – zprostředkovávají komunikaci s okolím
Mikroprocesor – provádí výpočetní operace

Vstupně-výstupní blok obsahuje obvody upravující komunikaci tak, aby mohla být zpracována po pevně velkých bitových řetězcích. Podle délky těchto pracovních řetězců se potom o mikrokontroléru mluví jako o 4, 8, 16 nebo 32 bitovém. Této délce jsou pak podřízeny další prvky jako registry, adresní sběrnice a datové sběrnice. Pozor, všechny tyto a další prvky v mikrokontroléru nemusí být stejné délky! Lze narazit na vyjímky. Typicky jsou to mikrokontroléry pracující naráz s osmi bity (8 bitová datová sběrnice), ale paměti jsou větší, než je možné obsáhnout osmibitovým kódem (musí mít větší adresní sběrnici a registry).

Registry jsou malé paměťové bloky, kterými disponuje mikroprocesor. Adresní sběrnice jsou propoje v obvodu sloužící k určení umístění dat v paměti. Datové sběrnice jsou propoje v obvodu sloužící k přenosu dat.

Je zvykem dělit architektury mikroprocesorů a mikrokontrolérů na tzv. Harvardskou a Von Neumannovu. Toto dělení vzniklo z potřeby mít standard pro stavbu počítače a usnadnit tak jeho vylepšování. Dnes jsou četné odchylky, ale hlavní je rozdíl v umístění pracovních dat a programu. Spojenou paměť pro oboje má Von Neumannova, oddělenou má Harvardská architektura.

2.2.4 Typy pamětí

Způsoby uchovávání dat prošly vývojem. První používané v mikrokontrolérech byly typu ROM a šly vyrábět pouze ve velkých sériích, protože vyžadovaly drahé výrobní procesy. Paměti typu ROM jsou pouze ke čtení, nelze tedy měnit programovou paměť kdykoliv se to hodí. Paměť byla nastavována pouze přímo při výrobě. Program musel být spolehlivě navrhnut před začátkem výroby a dělo se tak na modelech a emulátorech (simulátor architektury mikrokontroléru). Proto byly zavedeny paměti typu EPROM (EP znamená erasable programmable, mazatelné programovatelné). Vymazání programové paměti bylo prováděno krátkou expozicí UV záření. To usnadňovalo vývoj a aktualizaci

programu, protože testy a změny mohly probíhat na hotovém mikrokontroléru. Nevýhodou byla vyšší cena použitých součástí. Variantou EPROM byly OTP, paměti jedenkrát programovatelné. Z výroby byla u OTP vynechána okénka pro UV expozici. Zjednodušení znamenal vynález mazání pomocí elektrického náboje a tedy i pamětí typu EEPROM (EE znamená electrically erasable, elektricky mazatelné). Umožnilo vyrábět malé série, a ještě více usnadnilo nahrávání programu přímo na mikrokontroléru.

2.2.5 Spolehlivost

Mikrokontroléry jsou nasazovány v aplikacích, které jsou v provozu celé roky bez přestávky. Problémy týkající se výpadků zahrnují finanční ztráty (přerušení produkce), špatná dostupnost zařízení (satelit ve vesmíru), ohrožení života (signalizace železničního přejezdu), atp. Existuje řada funkcionalit, které zajišťují nebo usnadňují nápravu v případě chyby na mikrokontroléru, popřípadě na jeho periferiích:

„Hlídací pes“ – kontrolní obvod, který resetuje mikrokontrolér v případě, že se program přestane hlásit.

Nadbytečnost – části jsou v zařízení vícekrát a lze je za provozu mezi sebou přepínat.

Bezpečnostní provoz – komplexnější opatření, kdy systém omezí funkčnost na nejnutnější minimum, např. sníží otáčky motoru.

Bezpečné programování – při návrhu programu se využívají spolehlivé a zaběhnuté bloky kódu.

2.2.6 Strukturovaný přístup k programování

Je-li rozhodnuto o vybraných senzorech a řídící jednotce, zbývá náročná práce návrhu programu. Návrh má několik fází [8].

Analýza – formulace programem vykonávaných úkolů a sepsání specifikace.

Identifikace modulů – rozdělení celku na malé logické části, které jsou podkladem pro programování.

Implementace – převod na spustitelný kód ve vybraném programovacím jazyce. Programátor pracuje s osobním počítačem ve vývojovém prostředí.

Validace – program je testován na emulátorech a vývojových obvodech. Snažíme se ověřit, že cíle stanovené při analýze byly splněny.

Provoz – program je nasazen pomocí speciálního obvodu do daného zařízení. Na základě informace z provozu může docházet a většinou i dochází k dalším změnám v programu.

2.2.7 Programovací jazyky

Dá se vypozorovat vývoj programovacích jazyků směrem k většímu komfortu programátora a usnadňování jeho práce. Literatura dělí programovací jazyky na přibližně pět generací, pro mikrokontroléry se používají jazyky prvních tří. Platí, že čím nižší generace, tím takzvaně nižší jazyk, tedy blíže faktickým instrukcím hardware.

První generace – programování strojovým kódem, zapsané instrukce jsou přímo prováděny mikroprocesorem. Instrukce strojového jazyka se jeví jako binární kód délky podle typu mikroprocesoru. Neliší se od ostatních dat v paměti. Každý mikroprocesor má svoji sadu instrukcí, které musí programátor při tvorbě programu kódově zadávat. Programovací jazyky první generace jsou nejtěžší k ovládnutí a v praxi je málokdo používá.

Druhá generace – patří k nim tzv. sestavovací jazyk – assembler. Programátor pracuje se zkratkami reprezentujícími instrukce nebo jejich posloupnosti. Před spuštěním programu, musí proběhnout tzv. sestavení, což není nic jiného, než převod do strojového kódu. Assembler zpřehledňuje psaní programu, specifický pro daný mikrokontrolér a vyžaduje množství znalostí o hardware.

Třetí generace – příkazy se přibližují mluvené řeči, zavádí se bloková struktura programu, usnadňuje se práce se složenými proměnnými, jako jsou řetězce a pole. Je zde mohutný přesun zodpovědnosti za efektivitu kódu na software počítače. Do třetí generace patří mnoho jazyků, hlavním proudem jsou například COBOL, Fortran, BASIC a pozdější C.

2.2.8 Vývojové prostředí

Na většinu hlavních programovacích jazyků je vytvořeno integrované vývojové prostředí. Jeho hlavní osou je softwarová aplikace, která poskytuje programátorům množinu nástrojů pro vývoj software a zvyšuje tak jejich pracovní produktivitu. Zahrnuje editor zdrojového kódu se zvýrazňováním syntaxe, automatické doplňování kódu, kompilátor, sestavovací program, emulátor provádění programu (debugger), někdy i správu verzí software a obslužné nástroje repositářů (sdílených úložišť zdrojových kódů). Programátor může v jedné aplikaci kód jak psát, tak sestavovat a získat okamžitou zpětnou vazbu o chybách.

2.2.9 Řízení chodu programu

Cyklus – program prochází smyčkou, provádí rutiny postupně v pevně daném pořadí, neexistuje předbíhání

Přerušení – znamená, že provádění rutin může být spouštěno určitou událostí, např. příchozím bytem na sériové lince, nebo v určitých časových intervalech, které je možno měnit. Tento způsob má krátkou reakční dobu a je proto velmi vhodný pro řízení v reálném čase.

Kombinace cyklu a přerušení – je běžné, že úkoly méně náročné na rychlou reakci běží v cyklu a náročnější úlohy se spouští až když nastane přerušení. Některé mikrokontroléry umožňují vřadit pomocí obsluhy přerušení rutinu do hlavního cyklu. Tato rutina je potom zpracována cyklicky už bez nutnosti zásahu přerušení.

Nepreemptivní multitasking – podobá se řízení cyklem, Programátor definuje množinu úloh a každá je vykonávána dokud sama nezavolá uvolňovací rutinu.

Preemptivní multitasking – počítá s tím, že část kódu bude věnována přepínání mezi úlohami. Toto přepínání se děje v pravidelných intervalech a je ovládáno přerušením pomocí časovače. Potom mluvíme o přítomnosti operačního systému v mikrokontroléru. Jde o velmi komplexní úlohu a funkční řešení jsou dobrým prodejním artiklem. Jako příklad slouží Embedded Linux a Windows CE.

2.3 Navigační strategie

Protože robot v počáteční fázi vývoje nemá k dispozici žádný virtuální model okolního světa, jsou v této kapitole probrány přístupy k řízení robotu okolo překážek v bezprostředním dosahu senzorů.

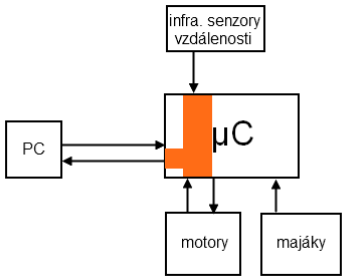
Reaktivní řízení – znamená, že robot reaguje přímo na stav prostředí. Nemusí přitom mít žádný reprezentační model prostředí a nevyvozuje z toho své další chování. Může naopak mít paměť s uloženým stavem předchozího kroku.

Procházka – nejjednodušší přístup vyhýbání se překážkám. Robot se pohybuje vpřed, dokud senzory nezaznamenají překážku, změní směr pohybu, vyhne se tak kolizi a pokračuje v pohybu vpřed.

Řízení vektorem – robot se pokouší objet překážku, za současného zachování přibližného směru pohybu. Vypočítává se vektor pohybu ze startu do cíle a přepočítává za chodu jeho aktualizovaná hodnota. Pokud narazí na překážku, uhne z trasy o stanovenou odchylku. Tato odchylka je nosným bodem různých známých strategií. V nejjednodušších řešeních je pouze pevně daná, v nejsložitějších ji propočítává několik počítačů k robotu připojených podle snímaných vlastností překážky. Když je překážka objeta, upraví se směr jízdy zpět podle aktuálního vektoru. [1]

3 POUŽITÉ ŘEŠENÍ

Robot bude vybaven elektromotorovou pohonnou jednotkou, několika infračervenými senzory vzdálenosti, mikrokontrolérovou řídící jednotku a do budoucna i systémem infračervených navigačních bodů v terénu (majáky). Prostory pohybu robotu budou místnosti s rovnými podlahami, bez prahů. Statickými překážkami bude běžný nábytek, krabice, zdi. Dynamickými budou jednotlivě se pohybující osoby. Prvním úkolem je zvládnout zpracování připojených infračervených senzorů, posílání stavových hlášení do počítače a ovládání motorů. Záběru tohoto úkolu odpovídá vybarvená oblast na blokovém schéma robotu.

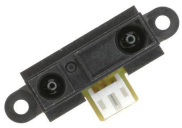


Obr. 6 Blokové schéma robotu B2.5

3.1 Zvolené senzory

Byl vybrán infračervený senzor Sharp GP2Y0A21. Detekční zóna tohoto senzoru je ve vzdálenosti 10-80 ± 0,5 cm. Výstupním signálem je napětí, které se mění v úměře ke vzdálenosti. Byla provedena linearizace funkce závislosti napětí na vzdálenosti.

Senzory jsou umístěny dva vpředu, dva na bocích a tři na podvozku.

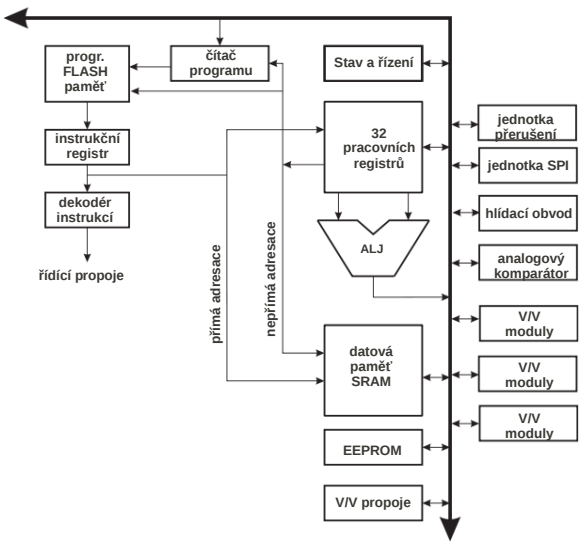


Obr. 7 Sharp GP2Y0A21, zdroj [12]

3.2 Zvolená řídící jednotka

Při konstrukci řídící jednotky bylo přihlédnuto kromě požadovaných vlastností robotu i ke zkušenostem domovského ústavu, dostupnosti součástí a vývojových nástrojů. Postupně byly vybrány mikrokontroléry ATmega8, ATmega16 a ATmega128. Jsou to oblíbené 8 bitové mikrokontroléry vyvíjené od roku 1996 v Norsku. Mají oddělenou paměť programovou a datovou, ale jediný adresní prostor, což brání jejich zařazení do čistě Harvardské architektury.

Atmega128 může být taktována až na 16MHz a provádět v každém tiku jednu instrukci. Jak lze dále vyčíst z obrázku, má 32 pracovních registrů připojených přímo k aritmetickologické jednotce (ALU). Ta může přistupovat ke dvěma registrům zároveň. [11]



Obr. 8 Blokové schéma mikrokontrolérů Atmega, zdroj [11]

Další vybrané vlastnosti:

128KB programová paměť typu FLASH, 4KB programová EEPROM, 4KB datová SRAM, 53 vstupně-výstupních vývodů, 4 časové čítače, pulsně šířkový modulátor, dva registry sériové linky, AD převodník s 8 kanály a 10 bitovým rozlišením, hlídací časovač, vnitřní oscilátor.

3.3 Zvolené vývojové prostředí

Bylo vyzkoušeno několik vývojových prostředí, kompilátorů, hardware programátorů a obslužného software, a to pro systémy Windows a Linux. Vzhledem k vybavení laboratoře bylo vybráno integrované vývojové prostředí AVRStudio s kompilátorem WinAVR od výrobce programovaného mikrokontroléru. Dále byl použit programátor a obslužný software od firmy PK Design.

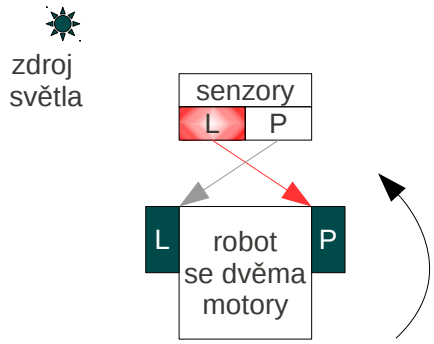
Prostředí AVRStudio umožňuje správu projektů pro snadné a přehledné vytváření rozsáhlých projektů , emulovat mikrokontroléry, překládat assembler do strojového kódu .

Kompilátor WinAVR je otevřený software, umí kompilovat z jazyka C a jde integrovat do AVRStudia.

Programátor a obslužný software od firmy PK Design jsou komerční produkty.

3.4 Zvolená navigační strategie

Na základě studií použitelných navigačních algoritmů bylo vybráno reaktivní řízení. Nejprve bez paměti vnitřních stavů a potom s pamětí. Na obrázku níže je typický příklad reaktivního řízení. Motor je sepnut, pokud na senzor na opačné straně robotu dopadá světlo.



Obr. 9 Typický příklad reaktivního řízení

4 APLIKACE VYTVOŘENÉHO ŘEŠENÍ

Ve spolupráci s vedoucím práce a kolegy v týmu byl navržen a zkonstruován autonomní mobilní robot B2.5. Na základě specifikace postupně vznikl navigační software, který prochází dalším vývojem.

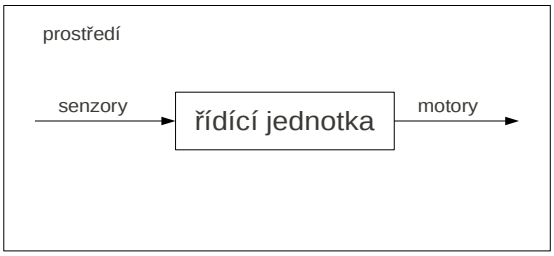
Obr. 10 Autonomní mobilní robot B2.5

4.1 Řídící algoritmus

První byla implementována náhodná procházka bez ukládání předchozích kroků. Po dosažení funkčních verzí infračervených majáků bylo řízení rozšířeno na cestu start-cíl s objížděním překážek.

4.1.1 Náhodná procházka

Tento typ navigace se řadí k čistě reaktivnímu, řídící jednotka neuchovává ve své paměti žádné ze svých vnitřních stavů. Robot snímá senzory nejbližší okolí a na základě toho vydává pokyny motorům.

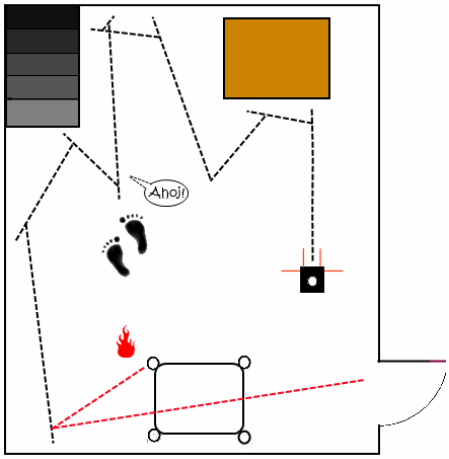


Obr. 11 Reaktivní přístup, zdroj [7]

Počáteční konfigurace je robot umístěný do volného prostoru místnosti (vizte obrázek níže). Pokud zaznamená svými senzory překážku, několik desítek milisekund sleduje, zda se překážka pohybuje. Pokud je překážka nepohyblivá, provede krátký pohyb vzad, otočí se do prostoru s větším volným prostorem a pokračuje v pohybu vpřed. Pokud je zaznamenána změna snímači namířenými k podlaze, je tato detekována jako díra. Reakce je podobná jako v případě nepohyblivé překážky. Nastane-li po kontaktu s překážkou změna ve vzdálenosti, je detekována pohyblivá překážka, předpokládá se, že před robotem prochází osoba. Pro fungování tohoto způsobu navigace musí být

odstraněny příliš úzké překážky, protože dva frontální senzory jsou montovány v určité vzdálenosti od sebe a je mezi nimi slepé místo.

Každý úkon zařídí posílání dohodnutého číselného kódu po sériové lince do počítače, je tak možné monitorovat chování robotu během jeho cesty.



Obr. 12 Ilustrace reaktivního řízení

4.1.2 Reaktivní řízení s vnitřní stavovou pamětí

Navigace tímto způsobem je závislá na zadání startovní pozice robotu, cílové pozice a místnosti jako mnohoúhelníku. Mnohoúhelník je zadán souřadnicemi krajních bodů. Robot přijímá signály z infračervených majáků. Na základě zadaných údajů a informaci o poloze vůči majákům je pak vypočten vektor pohybu k cíli. Tento vektor je určen souřadnicemi X, Y a φ, a v pravidelných intervalech se aktualizuje.

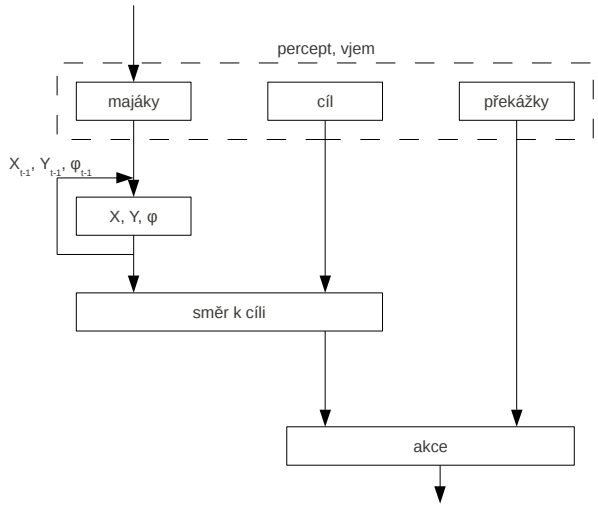
Jeho bezprostřední reakce je možné nastínit v pseudokódu následovně: [7]

```
state = c0;
while 1 do
        percept = see();
        state = next_state(state, percept);
        action = select(percept, state);
        if action != null then do (action)
end

function select(percept, state):TAction
        action:=null;
        switch(state)
                case s0: action = aca
                        switch(percept)
                        case p1: action = aca
                        case p2: action = aca
                        ..
                        case pm: action = aca
                case s1: action = acb
                ..
                case sn: action = acx
        return(action)
end
```

Funkce see() zobrazuje vjem v aktuálním stavu prostředí, funkce do(action) znamená provedení akce „action“ robotem. Systém nemá vnitřní model prostředí, nicméně jeho stav závisí na minulých stavech systému – jedná se o systém závislý na sekvenci předchozích podnětů a vstupů. Přemisťování k cíli je závislé na těchto stavech.

Přístup je konkretizován na dalším obrázku:



Obr. 13 Reaktivní navigace robotu B2.5

Robot vnímá pozici majáků rozmístěných v prostoru, snímá senzoricky překážky a má informaci o souřadnicích cíle. Dále je známa poloha a natočení robotu a uložen stav z předchozího kroku. Z těchto údajů se propočítává úprava cesty k cíli a ta má vliv na okamžitý směr jízdy.

4.2 Komunikační protokol

Společně s robotem vzniká ovládací software pro osobní počítač, který usnadňuje zadávání inicializačních dat. Příkazy budou robotu zasílány po sériové lince, bezdrátově přemostěné pomocí technologie Bluetooth.

Pro správné vyhodnocení informací příchozích do robotu, byl navržen systém pevných datových struktur, 30 byte dlouhých paketů. Paket sestavený ovládacím programem je uvozen startovací sekvencí dvou byte, každý s hexadecimální hodnotou 0xFF. Následuje identifikační byte, jenž rozlišuje typ informací v paketu. Jsou zavedeny tři typy paketů: s informací start-cíl, polohách majáků a hranice oblasti ve které se má robot pohybovat. Poslední dva byte jsou rezervovány pro kontrolní součet.

Na obrázku je příklad paketu s informací o startovní pozici robotu. Barevné bloky jsou jednotlivé byte paketu, černé jsou oddělovače. Význam podle pozice byte je následující:

1., 2. startovní
3. identifikace paketu
4., 5. souřadnice X cíle
6., 7. souřadnice Y cíle
8., 9. souřadnice X startovní
10., 11. souřadnice Y startovní
12., 13. úhel φ startovní natočení robotu
14..28: prázdné
29., 30. kontrolní součet

Obr. 14 Grafická reprezentace paketu

Pro příjem inicializačních dat se robot po zapnutí přepne do stavu příjmu paketů. Z počítače přichází postupně data, řídící jednotka odpovídá sekvencí s významem buď potvrzujícím přijetí, nebo vyžadujícím nové vysílání. Poslední v inicializační komunikaci je sekvence z řídícího software, značící konec přenosu. Řídící jednotka robotu přechází z inicializační fáze do fáze provozní a vydává se k cíli.

5 ZÁVĚR

S pomocí vedoucího práce a týmu kolegů vznikl autonomní robot B2.5, jenž se dále vyvíjí, zlepšuje se jeho navigační schopnost a přibývají funkce. Při jeho vývoji se pracovalo se zkušenostmi z předchozích verzí robotu B, kteří se úspěšně účastnili soutěží v jízdě podle map. Souběžně se pracuje na robotu B3, který převezme některé technologie z B2.5 a má potenciál pro praktickou aplikaci.

Vývoj vyžadoval intenzivní práci a studium nad rámec předmětů vyučovaných domovským ústavem. Vznikl funkční tým kolegů vedených školitelem, který musel práci koordinovat, rozdělovat úkoly a kompetence. Byla to cenná zkušenost blízká praktickému fungování týmu spolupracovníků s projektovým manažerem.

Pro splnění úkolů pokrytých touto prací bylo potřeba proniknout do programování mikrokontrolérů. To vyžaduje studium materiálů často psaných v angličtině. I když se přistupovalo s určitými předchozími znalostmi o programování v jazyce C, objevily se neúspěchy, ale tyto byly překonány díky konzultacím s vedoucím práce. Zvolený jazyk je výhodný, protože umožňuje programovat různé verze mikrokontroléru s minimem změn v kódu, nebo i úplně jiné mikrokontroléry s patřičným kompilátorem. Vyzkoušeno bylo několik improvizovaných řešení programu, než se dospělo k optimální cestě vývoje tak složitého systému, jako je robot. Utřídit myšlenky se podařilo s pomocí grafických diagramů. Vizuální přístup k rozkladu problémů je zřejmě výhodnější a silnější, než prostá soupiska.

Zajištění převodu programu do mikrokontroléru vyžadovalo konstrukci programátoru USBAsp podle návodu. Touto cestou bylo nasbíráno množství zkušeností s výrobou desek pájených spojů nažehlováním fólií a expozicí kuprextitu UV zářením. S úspěchem lze takto vyrábět senzorické a řídící obvody osazené minimálním množstvím elektronických součástek.

Aby bylo vůbec možné navrhnout řídící software pro autonomní mobilní robot, bylo nutné ovládnout pestrou směsici komunikačních protokolů a standardů, jimiž různé prvky komunikují. Praktické nasazení komunikace standardu RS-232 bylo ověřováním si předchozích teoretických znalostí, stejně tak funkce AD komparátorů. Zejména RS-232 se ukázalo jako velmi spolehlivé využitelné pro technickou praxi, má zkrátka svoje využití i v době, kdy vládnou USB rozhraní.

Zadávání inicializačních dat usnadňuje software pro osobní počítač. Tento komunikuje přes sériovou linku přemostěnou technologií Bluetooth pomocí dohodnutých standardních paketů.

Navigační software pro procházku místností ukázal na limity celého zařízení a napověděl, jak bude muset být uspořádáno prostředí, kde se bude robot pohybovat. Je schopen si poradit s většinou fyzických překážek v místnosti laboratoře. Připravované další verze robotu řady B budou potřebovat dokonalejší soustavu senzorů, aby bylo dosaženo uspokojivých výsledků.

SEZNAM POUŽITÉ LITERATURY

[1] EVERETT, H. R. Sensors for mobile robots: theory and application, A K Peters, 1995, ISBN 1-56881-048-2

[2] LITTLE, Cheryl. The Intelligent Vehicle Initiative: Advancing "Human-Centered" Smart Vehicles [online]. 1997, září. [cit. 7. května 2010]. Dostupné z: <http://www.tfhrc.gov/pubrds/pr97-10/p18.htm>

[3] Mediální repositář Wikimedia Commons. Wikimedia Foundation. [20. května 2010]. Dostupné z: <http://commons.wikimedia.org>

[4] NOVÁK, Petr. MOBILNÍ ROBOTY – pohony, senzory, řízení. BEN – technická literatura. 2005. 243 s. ISBN 80-7300-141-1

[5] GOTTSCHALK, Mark A., Sensors make cars smarter, Western Technical Editor – Design News, [online], 1997, 5. říjen. [cit. 26. května 2010]. Dostupné z: <http://www.designnews.com/article/13464-Sensors_make_cars_smarter.php>

[6] KUBÍK, A. Inteligentní agenty. Computer Press. 2004. 280 s. ISBN: 80-251-0323-4

[7] ZBOŘIL, František. Podklady k přednáškám kurzu AGS, Agentní systémy, základní architektury, jejich modely a realizace. 2005, 2006. [cit. 20. května 2010]. Dostupné z: <www.fit.vutbr.cz/~zborilf/study/AGS/AGS02_Zakladni_modely.pdf>

[8] RÁČEK, Jaroslav. Strukturovaná analýza systému. 2000. 80 s. ISBN 80-210-4190-0

[9] BISHOP, Owen. Robot Builder's Cookbook. Newnes. 2007. 373 s. ISBN: 978-0-7506-6556-8

[10] JONES, L. Joseph, SEIGNER, A. Bruce, FLYNN, M. Anita. Mobile Robots Inspiration to Implementation. PETERS, A.K. 1997. ISBN: 1-56881-097-0

[11] Atmega128. Datasheet [online]. 2008. [cit. 20. května 2010]. Dostupné z <www.atmel.com/atmel/acrobat/doc2467.pdf>

[12] SHARP GP2Y0A21. Datasheet [online]. 2008. [cit. 26. května 2010]. Dostupné z: <http://datasheet4u.com/>