

Mikropočítačové ovládání mobilního robotnického systému

Microprocessor-based Kontrol of Mobile Robot

Tomáš Král

Bakalářská práce
2009



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2008/2009

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš KRÁL**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Mikropočítačové ovládání mobilního robotického systému**

Zásady pro vypracování:

1. Vypracujte literární rešerši na téma mobilních robotických systémů.
2. Analyzujte možnosti robota e-puck.
3. Navrhněte ukázkovou aplikaci využití robota e-puck.
4. Demonstrujte navrženou aplikaci.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. OWIE, Choset. HUTCHINSON, Seth. LYNCH, Kevin. KANTOR, George. BURGARD, Wolfram. KAVRAKI, Lydia. THRUN, Sebastian. Principles of Robot Motion: Theory, Algorithmus, and Implementation. MIT Press, 2005. 565s. ISBN 0262033275, 9780262033275.
2. DUDEK, Gregory. RICHARD, Michael. JENKIN, MacLean. JENKIN, Michael. Computational Principles of Mobile Robotics. Cambridge University Press, 2000. 280s. ISBN 0521568765, 9780521568760.
3. THRUN, Sebastian. BURGARD, Wolfram. FOX, Dieter. Probabilistic Robotics. MIT Press, 2005. 667s. ISBN 0262201623, 9780262201629.

Vedoucí bakalářské práce:

Ing. Erik Král

Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce:

20. února 2009

Termín odevzdání bakalářské práce:

1. června 2009

Ve Zlíně dne 13. února 2009


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem diplomové práce je poskytnout přehledný návod pro programování mobilního robota e-puck v programovém prostředí MPLAB s kompilátorem C30. Teoretická část práce se zaměřuje na popis senzorických a lokomočních ústrojí mobilních servisních robotů a na popis programových prostředí, ve kterých lze mobilního robota e-puck programovat. Praktická část práce obsahuje stručný návod zabývající se založením projektu v programu MPLAB, kompilací projektu a nahráním kompilací získaného programu do mobilního servisního robota e-puck pomocí programu Tiny Bootloader. Dále se praktická část zabývá popisem funkcí vybraných knihoven používaných při tvorbě aplikací pro mobilního robota e-puck.

Klíčová slova: e-puck, mobilní robot, C++

ABSTRACT

The purpose of bachelor thesis was to create transparent instruction for programming mobile robot e-puck in MPLAM program with C30 compiler. Theoretical part was focused on description of sensorial and locomotive mobile mechanism of service robots and on description of programs in which can be e-puck robot programmed. Practical part contains brief guide of creation project in MPLAB program, of project compilation and about uploading the program in mobile service robot e-puck with help of Tiny Bootloader program. In next part there is description of functions from selected libraries, which are used to creation of applications for mobile robot e-puck.

Keywords: e-puck, mobile robot, C++

Poděkování, motto:

Na tomto místě bych rád poděkoval vedoucímu mé bakalářské práce Ing. Eriku Králi za odbornou pomoc a vedení při řešení problémů s touto prací.

(Isaac Asimov, The caves of steel, Galaxy No. 13, 1950.)

1. Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby člověku bylo ublíženo.
2. Robot musí uposlechnout příkazu člověka, kromě případu, kdy jsou tyto příkazy v rozporu s prvním zákonem.
3. Robot musí chránit sám sebe před zničením, kromě případu, kdy je tato ochrana v rozporu s prvním nebo druhým zákonem.

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve

.....

Zlíně

Podpis diplomanta

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 MOBILNÍ ROBOT	10
1.1 SENZORICKÝ SYSTÉM	10
1.1.1 Vnitřní snímače	11
1.1.2 Vnější snímače	11
1.2 ROZDĚLENÍ ROBOTŮ PODLE PODVOZKU	11
1.2.1 Rozdělení podle prostředí.....	11
1.2.2 Rozdělení podle lokomočního ústrojí.....	14
1.2.3 Mobilní roboty na kolovém podvozku	15
1.2.3.1 Tříkolové a čtyřkolové podvozky	15
1.2.3.2 Šestikolové podvozky	17
1.2.3.3 Speciální kolové podvozky pro mobilní asistenční roboty	18
1.2.4 Pásové roboty	18
1.2.5 Roboty se všesměrovými koly	19
1.2.6 Kráčeující roboty	19
1.2.6.1 Dvounohé kráčeující roboty	20
1.2.6.2 Čtyřnohé kráčeující roboty	21
1.2.6.3 Šestinohé kráčeující roboty	21
1.2.6.4 Osminohé kráčeující roboty.....	22
1.2.7 Mikroroboty	23
1.2.8 Plazivé mobilní robotické systémy	24
1.2.9 Speciální mobilní robotické systémy	25
1.2.9.1 Šplhající mobilní robotické systémy	25
1.2.9.2 Plovoucí robotické systémy.....	26
1.3 VÝVOJOVÁ PROSTŘEDÍ	27
1.3.1 MPLAB C30.....	27
1.3.2 Pikilab.....	27
1.3.3 Matlab	27
II PRAKTICKÁ ČÁST.....	28
2 ZPROVOZNĚNÍ MOBILNÍHO ROBOTY E-PUCK	29
2.1 ROZBLIKÁNÍ LED DIODY (HELLO WORLD)	29
2.1.1 Vytvoření projektu.....	29
2.1.1.1 Výběr zařízení.....	30
2.1.1.2 Výběr pracovního balíčku.....	30
2.1.1.3 Jméno projektu	31
2.1.1.4 Přidat do projektu existující soubory	32
2.1.2 Zkompilování projektu	34
2.1.2.1 Nastavení kompilačních parametrů	34
2.1.2.2 Samotná kompilace	36
2.1.3 Naprogramování robota e-puck.....	37
2.1.3.1 Spárování robota e-puck přes Bluetooth	37
2.1.3.2 Flashnutí robota e-puck.....	40

2.2	KNIHOVNY.....	41
2.2.1	e_init_ports.....	41
2.2.2	e_led.....	41
2.2.3	e_motors.....	42
2.2.4	e_sound.....	42
2.2.5	P30f6014A.gld.....	42
ZÁVĚR		43
ZÁVĚR V ANGLIČTINĚ		44
SEZNAM POUŽITÉ LITERATURY		45
SEZNAM OBRÁZKŮ		46
SEZNAM PŘÍLOH		48

ÚVOD

Když v roce 1920 Josef Čapek vytvořil slovo robot a jeho bratr Karel ho ve svém románu R.U.R. popsal jako nenáročný, výkonný a myslící stroj, který nahrazuje lidskou práci, asi netušili, jak brzy se tato myšlenka stane skutečností. Robotické systémy od těch nejjednodušších až po ty nejsložitější během minulého století pronikly nejen do velkých průmyslových firem a výzkumných laboratoří, ale i do domácností a běžného života každého z nás. Nedá se předpokládat, že by se tento trend zastavil, robotika má proto před sebou slibnou perspektivu. S ohledem na tuto skutečnost roste i význam studijních prostředků v oblasti robotiky. Mezi tyto prostředky patří i mobilní robot e-puck, pro který je v rámci této diplomové práce sepsán manuál a ukázkový program. Jako nejvhodnější prostředek k psaní aplikací pro mobilního robota e-puck byl zvolen program MPLAB s kompilátorem C30 který překládá z programovacího jazyka C++ do zdrojového kódu pro mikrokontroler P30F6014. Program samotný se sice píše v Jazyce C++, ale programovací jazyk assembleru je nutno znát také, neboť některé části knihoven jsou napsány pouze v assembleru a pro pochopení jejich funkce je tato znalost nezbytná. Pro přenos vytvořené aplikace do mobilního robota e-puck, přes virtuální sériový port vytvořený pomocí Bluetooth, byl zvolen program Tiny Bootloader. Teoretická část práce se zabývá rozdělením mobilních robotů podle lokomočního ústrojí na roboty na kolovém podvozku, pásovém podvozku a na krácející roboty. Dále jsou tyto podvozky dále rozděleny a je popsána jejich funkce a využití.

I. TEORETICKÁ ČÁST

1 MOBILNÍ ROBOT

Výrazný rozvoj výpočetní techniky, způsobený rozvojem elektrotechniky v posledních dvou desetiletích, má velice pozitivní vliv na rozvoj robotiky. Robotika je závislá na možnosti zpracovávat údaje ze snímačů, tyto údaje vyhodnotit a vydat příslušné povely servomotorům, a to vše v reálném čase. Robotem rozumíme integrovaný, počítačem řízený systém, který je cílově orientované interakce s prostředím v souladu s instrukcemi operátora.

Autonomní roboty jsou roboty, provádějící požadované úkoly bez nepřetržitého lidského vedení. Různé činnosti vyžadují roboty s různým stupněm autonomie. Vysoký stupeň autonomie je zvláště žádoucí v oblastech, jako je například vesmírný průzkum.

System, který zabezpečuje působení robota na okolní prostředí a pohyb robota se nazývá motorický systém. System, který přímá informace z okolního prostředí se nazývá senzorický systém.

1.1 Senzorický systém

Všechny vstupní údaje, jako jsou údaje o pohybu robota, informace o prostředí, a rozhodování o vykonání jednotlivých operací zpracovává a vyhodnocuje řídicí systém robota. Pro zabezpečení správné činnosti robota je též zapotřebí, aby měl pod kontrolou svůj vlastní pohyb.

Informace získané jednotlivými snímači senzorického systému, zpracované v řídicí jednotce robota, umožňují vytvořit vnitřní model okolního prostředí a na jeho základě řídicí jednotka koordinuje vykonávání jednotlivých operací podle naprogramované úlohy. Podle prostorového vztahu k podnětům dělíme snímače na dva druhy:

- Vnitřní
- Vnější

1.1.1 Vnitřní snímače

Snímače vnitřní zpětné vazby se používají ke sledování vlastního pohonného, případně jiného manipulačního systému, například na určení přesné polohy pohonného servomotoru.

1.1.2 Vnější snímače

Snímače vnější zpětné vazby se používají ke zkoumání předmětů v robotově okolí, zjištění jejich tvaru, velikosti, případně i barvy nebo struktury. Používají se tedy k navigaci robota, případně ke komunikaci s operátorem. Vnější snímače můžeme dále rozdělit na:

- Radiové (radiolokátor, GPS)
- Ultrazvukové (radar)
- Zvukové (mikrofon)
- Optické (infračervené, laserové, fotoaparáty)
- Tlakové (tlačítko)

Pro více informací doporučuji knihu [7]

1.2 Rozdělení robotů podle podvozku

Mobilita je specifickou vlastností robota pro roboty s podvozkem, nebo jiným systémem, který umožňuje pohyb robota.

Současné trendy světového vývoje v oblasti servisních a mobilních robotů ukazují na velice široké spektrum možnosti jejich uplatnění. Na rozdíl od průmyslových robotů nacházejí mobilní a servisní roboty uplatnění i v netradičních oblastech. Je to již ze samotné definice robota. Roboty jsou definovány jako technická zařízení, která se podílejí na nevýrobních činnostech. V poslední době se velice dobře uplatňují v oblasti služeb a asistenčních technologií.

1.2.1 Rozdělení podle prostředí

Základním faktorem, který má vliv na konstrukční parametry a chování servisního robota, je prostředí, ve kterém se robot bude pohybovat. Zde lze uplatnit dva základní typy prostředí:

- Venkovní (Outdoor)
- Vnitřní (Indor)

Pro venkovní prostředí je charakteristické, že se robot bude pohybovat mimo budovy na různě členitém terénu, jako jsou lesy, pole, krátery, městská zástavba. Tuto oblast terénu je také možno rozšířit o pohyb v zamořeném prostředí, podmořský výzkum, průzkum cizích planet a jiných vesmírných těles. Jedná se tedy o terén složitější než terén vnitřní.

Pro vnitřní prostředí je charakteristické, že se bude robot pohybovat pouze v budovách, rozsáhlých nádržích, vzduchovodech, potrubích a podobných prostorách. Jedná se tedy o terény jednodušší povahy, než jsou terény venkovní.

Obecně lze říci, že mobilní roboty jsou určeny pro vykonávání úloh v částečně známém nebo zcela neznámém prostředí s neustále se měnícími scénami. Ve většině aplikací se dnes požaduje zcela, nebo alespoň částečně autonomní chování. Aby robot mohl pracovat autonomně, musí využívat propracovaných rozhodovacích procesů a adaptabilního nebo supervizorového řízení. Roboty s autonomním řízením jsou tedy technická zařízení schopná provádět, interpretovat, ale hlavně plánovat zadanou úlohu na základě vykonávání programu řídicím systémem a informací získaných ze senzorického systému a systému zpětné vazby.

Veškerá činnost mobilního robota je ovlivněna zejména polohovacími a orientačními ústrojími, efektory a v neposlední řadě také podvozkovou částí. Efektory a orientační ústrojí bývají přizpůsobeny pro vykonávání požadovaných úkonů. Na rozdíl od robotů průmyslových se jedná převážně o práci s křehkými předměty. Efektory proto bývají konstrukčně řešeny trochu jiným způsobem a to hlavně tím, že obsahují větší množství senzorů než roboty v průmyslové oblasti. Jedná se především o snímače zjišťující sílu úchopu, tyto snímače nazýváme snímači taktilními. Konstrukce efektoru se velmi často přibližuje konstrukci lidské ruky několika prsty. Nejčastěji můžeme u takovýchto efektorů vidět dvěma až třemi prsty proti jednomu palci. Aby robot nebyl stacionární, ale mobilní, je zapotřebí, aby byl obdařen efektozem, umožňujícím robotu přesun jeho samého. Roboty jsou konstruovány s takovým podvozkem, který nejlépe vyhovuje prostředí a pracovní náplni robotu. Jedná-li se o činnost nutně vyžadující velkou manévrovatelnost, je například zvolen podvozek se všesměrovými koly nebo s více stupni volnosti, je-li vyžadován

rovnoměrný chod robotu, například při transportu nebezpečných předmětů, vzorků, pacientů v nemocnici, volí se podvozek obsahující větší množství kol, popřípadě podvozek pásový.

Z hlediska konstrukce podvozků můžeme rozdělit mobilní roboty do několika základních skupin:

- Kolové
 - hybridní (obsahující kola i pásy)
- Pásové
 - hybridní (obsahující kola i pásy)
- Kráčejíci
 - kráčejíci
 - plazivé
 - skákací
 - šplhací
- Ostatní
 - Plavací
 - Létací

1.2.2 Rozdělení podle lokomočního ústrojí

Pod pojmem lokomoční ústrojí rozumíme technický prostředek umožňující pohyb robotu, v našem případě podvozek. Konstrukční řešení podvozku robotu je závislé na různých principech a řešeních jednotlivých detailů v závislosti na činnosti robotu a vlastnosti prostředí, ve kterém bude robot tuto svou předem definovanou činnost vykonávat.

Mobilní servisní roboty mohou být dvojího typu:

- Drátový (lokálně řízený)
- Terminálový (globálně řízený)

Podle tohoto rozdělení je nutné připravit matematický model pro dané prostředí, ve kterém se robot bude pohybovat, a pro čas, po který bude robot tuto činnost provádět. U moderních mobilních robotů je již samozřejmostí schopnost vyhýbat se náhodným překážkám, přemístit se do určitého bodu s co největší přesností, počítačová analýza prostředí v okolí robotu, kopírování terénu, a mnohé další. Ke splnění všech těchto požadavků je zapotřebí, aby robotův podvozek měl dobré manévrovací schopnosti, to znamená, aby byl schopen otáčet se na místě kolem svislé osy, zatáčet ve velmi ostrém úhlu apod. Pro různé oblasti použití robotu jsou charakteristické různé typy podvozku robotu.

Co se týče oborů, v nichž našly roboty využití, mimo strojírenství, kde je snad největší zastoupení robotů a robotických systémů, lze uvést například zdravotnictví, stavebnictví, zemědělství, lesnictví, armáda, a další. Specifickou oblastí, z hlediska požadavků na konstrukční parametry, je průzkum pod vodní hladinou. Další velmi specifickou oblastí, z hlediska požadavků na konstrukční parametry mobilních robotů, je průzkum povrchu vesmírných těles, například cizí planety, přirozené družice, ale také průzkum samotného kosmického prostoru. Každá z těchto výše zmiňovaných oblastí má svá vlastní specifika a uplatnění v nich mohou najít mobilní roboty s různým lokomočním ústrojím.

1.2.3 Mobilní roboty na kolovém podvozku

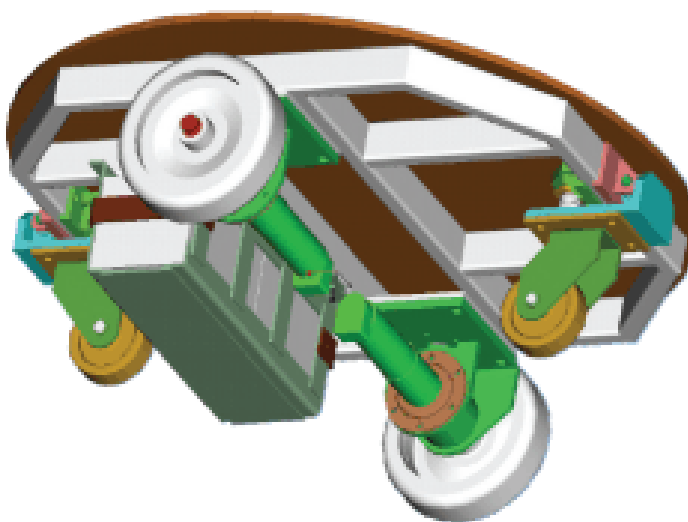
Roboty s kolovými podvozky se vyznačují dobrou schopností manévrovat a jsou také vhodné k překonávání překážek. Nejčastější verze kolových podvozků jsou podvozky tříkolové, čtyřkolové nebo šestikolové. U podvozků s méně než třemi koly, je třeba řešit problém s udržení rovnováhy, proto se téměř nepoužívají. Podvozky s více než šesti koly se také příliš nepoužívají.

Každý z výše zmiňovaných kolových podvozků má svá specifika, a proto mobilní servisní roboty s různými lokomočními ústrojími nacházejí uplatnění v různých prostředích a pro různé činnosti.

1.2.3.1 Tříkolové a čtyřkolové podvozky

Tříkolové a čtyřkolové mobilní servisní roboty lze rozdělit na diferenčně řízené roboty, roboty s více stupni volnosti, roboty řízené Ackermanovým způsobem a synchronně řízené roboty.

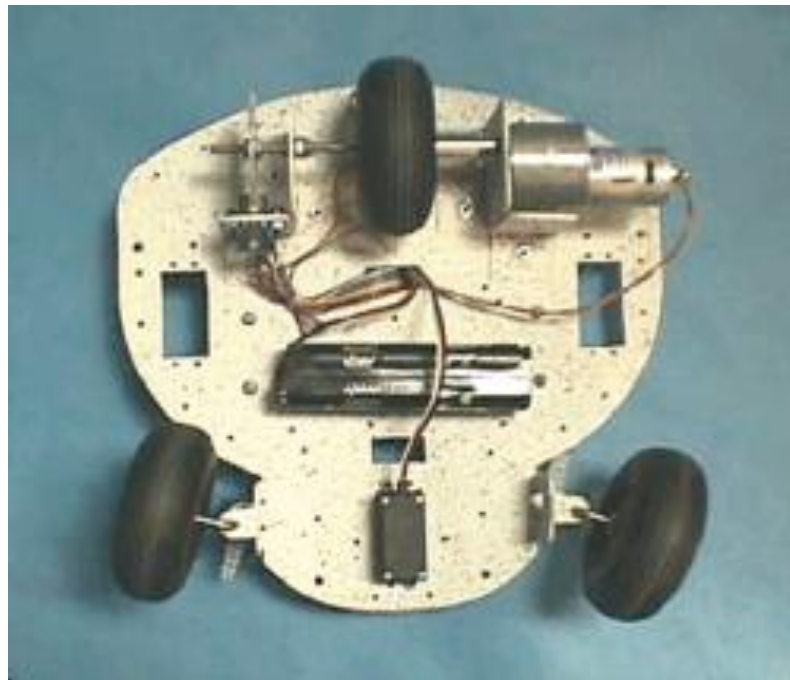
- **Diferenčně řízené roboty** se vyznačují tím, že mají nezávisle poháněná dvě kola a vpředu (popř. vzadu) volně otočné nepoháněné směrové kolo (popř. kola).



Obrázek 1 – Diferenčně řízený robot

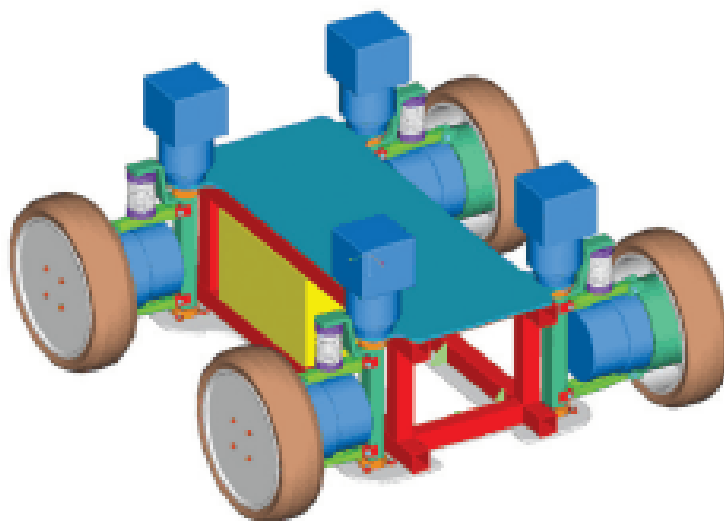
- **Tříkolový robot řízený Ackermanovým způsobem** má jedno poháněné kolo a dvě kola řiditelná nebo dvě poháněná a jedno řiditelné. Pro čtyřkolový robot řízený tímto

způsobem platí, že má dvě poháněná kola, která musí být vybavena mechanickým nebo elektrickým diferenciálem, a vpředu (popř. vzadu) otočná nepoháněná kola.



Obrázek 2 – Tříkolový robot řízený Ackermanovým způsobem

- **Synchronně řízené roboty** jsou opatřeny třemi nebo čtyřmi koly, přičemž všechna jsou poháněna a řízena tak, že mají navzájem stále stejné natočení a rychlost. Pro natáčení kol se používá společný hnací řetěz a jeden společný řetěz k pohonu kol. Mobilní roboty s těmito typy podvozků nacházejí uplatnění ve venkovním prostředí. [1]



Obrázek 3 – Synchronně řízený robot

1.2.3.2 Šestikolové podvozky

Šestikolové mobilní roboty lze zařadit do zvláštní skupiny, většinou jsou totiž stejně jako pásové podvozky řízeny smykem, nebo mají řízený přední a zadní pár kol. Takto konstruované podvozky se pro vnitřní prostředí v podstatě nepoužívají, jejich konstrukce je předurčuje pro použití ve venkovním prostředí, protože jsou schopny zdolávat členitý terén a nerovnosti. Jelikož jsou sestrojovány pro překonávání terénních překážek a vůbec pohybu v náročném prostředí, vyžadují i pohon o odpovídajícím výkonu. Technickou a technologickou převahu v tomto typu pohonu má zcela jednoznačně americká společnost pro výzkum vesmíru NASA, která již řadu let vyvíjí šestikolové mobilní roboty pro pohyb na cizích planetách. V současnosti jsou mobilní servisní roboty na šestikolovém podvozku využívány také v chemickém nebo jaderném průmyslu a v neposlední řadě i v těžebním průmyslu.



Obrázek 4 – Mobilní sonda Sojourner (NASA)

1.2.3.3 Speciální kolové podvozky pro mobilní asistenční roboty

Do skupiny kolových podvozků pro asistenční roboty se zahrnují také podvozky, které využívají speciální kola, jako jsou všesměrová kola, Weinsteinova kola a článkové pojezdy, kola MaxWheel, šnekové podvozky nebo kola robotů zdolávajících svislé překážky. [2]

Nyní se ještě ve stručnosti zmíním o některých dalších konstrukčních řešení robotů, využívajících ke svému pohybu prostředky jiné, než kola.

1.2.4 Pásové roboty

Pásové roboty nacházejí svá uplatnění v celé řadě aplikací, díky své speciální pásové konstrukci využívající řízení tzv. smykem.

Bývají využívány jako podvozky u různých druhů tanků, buldozerů, vojenských nosičů a pro mnohá další zařízení sloužící pro pohyb ve špatně schůdném terénu – převážně se jedná tedy o venkovní aplikace. Ve vnitřním prostředí se uplatňují zejména pro jízdu do a nebo ze chodů a pro aplikace s potřebou velmi vysoké stability. Jelikož řízení smykovým způsobem není pro autonomní roboty v žádném případě přesný, používají se pásové roboty především ve spojení s teleoperátorem.

Nejčastěji tedy nalezneme pásové roboty v těchto odvětvích:

- jaderný průmysl
- stavebnictví
- vojenské a policejní
- likvidace požárů
- práce s nebezpečným odpadem
- průzkum neznámého terénu
- asistenční technologie a rehabilitační robotika



Obrázek 5 – Čtyřpásový mobilní robot

1.2.5 Roboty se všesměrovými koly

Roboty vybaveny všesměrovými koly nejsou vhodné ke zdolávání překážek, nehodí se proto příliš k vykonávání venkovních aplikací. Uplatnění tedy nalézají při plnění aplikací uvnitř budov, jako jsou:

- ostraha
- vyhledávání nebezpečných předmětů
- monitorování stavu povrchů různých ploch
- monitorování prostorů nedostupných pro člověka

1.2.6 Kráčející roboty

Kráčející roboty lze rozdělit podle několika hledisek:

- dle počtu nohou - 2, 4, 5, 6, 8 a vícenohé roboty
- dle provedení nohy - typ savec, typ hmyz
- dle kinematických vazeb - rotační a translační

Nejčastějším provedením kráčejších robotů jsou provedeny jako kráčejší plošiny na čtyřech nebo šesti nohou, a jako humanoidní na dvou nohou. Na konstrukci téměř všech dvounohých robotů můžeme vidět snahu napodobit člověka a stavbu jeho těla.

1.2.6.1 Dvounohé kráčejší roboty

Dvounohé humanoidní roboty, na rozdíl od kolových a pásových robotů, jsou schopné překonávat překážky, jako jsou schody. V současné době jsou již roboty vybaveny natolik schopným hardwarem a softwarem, že nemají potíže s plynulou hladkou chůzí a jejich stabilita a manévrovatelnost je také velmi vysoká.



Obrázek 6 – Mobilní kráčejší robot Honda Asimo P3

1.2.6.2 Čtyřnohé kráčejíci roboty

Konstrukce čtyřnohých kráčejících robotů je většinou inspirována anatomii živočichů. Nejčastěji můžeme vidět konstrukce napodobující konstrukce typu „savec“ nebo typu „hmyz“. Toto pojmenování je rámcové a je provedeno na základě konstrukce končetiny a nikoliv podle počtu končetin.

Tyto mechanismy jsou staticky stabilní, ale v okamžiku, kdy jedna noha není v kontaktu s povrchem, se těžiště robotu nachází nad hranou trojúhelníka z bodu kontaktu ostatních nohou s povrchem nebo v blízkosti této hrany. V důsledku dynamického účinku nebo náhodné překážky či prohlubně může dojít k převážení robota na nohu vykonávající krok. Čtyřnohé kráčejíci roboty jsou proto vybavovány větším počtem senzorů než roboty s šesti a více nohama. [4]



Obrázek 7 – Robot savec jménem BigDog

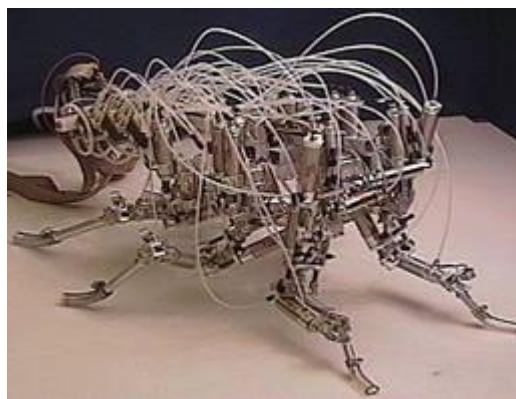
1.2.6.3 Šestinohé kráčejíci roboty

Šestinohé roboty jsou jistým druhem kompromisu mezi roboty čtyřnohými, které jsou jednodušší jak po hardwarové tak softwarové stránce, a osminohými s hladce plynulým

pohybem. Konstrukce šestinožých robotů bývá nejčastěji inspirována stavbou hmyzího těla, které v sobě skrývá velký potenciál.

Tato skupina kráčejších robotů je zjevně nejpočetnější. Některé roboty s této kategorie se v případě převrácení na záda dokáží otočit zpět na nohy a pokračovat dále v chůzi.

Šestinohé roboty splňují podmínky stability dokonale, protože vždy nejméně tři nohy zůstávají při chůzi stále v kontaktu s povrchem. Těžiště robotu se pak stále nachází uvnitř trojúhelníku vytvořeného body dotyku. Proto je chůze robotu vysoce stabilní a adaptabilní na povahu terénu a kráčejší robot je velmi odolný vůči převrnutí. Je výrazně potlačen vliv dynamických účinků, takže není třeba znát v každém okamžiku veškeré informace o členech klimatického řetězce. Výrazně tak klesá množství potřebných senzorů a zjednodušuje se řídicí systém. [4]



Obrázek 8 – Hmyzí robot CWRU III

1.2.6.4 Osmínohé kráčejší roboty

V této skupině se vyskytují kromě konstrukcí podobných šestinožým kráčejším mechanismům pro chůzi po zemském povrchu i roboty pro pohyby potrubím a po kolmých stěnách. Osmínohá koncepce je pro tyto účely nejvhodnější, protože roboty se mohou pomocí čtyř noh dostatečně stabilně uchytit a zajistit pevnou vazbu s povrchem, přičemž ostatní čtyři nohy vykonávají krok vpřed. V případě, že jedna noha ztratí kontakt s povrchem, musí zbylé tři nohy udržet robota na místě. Při chůzi po vodorovných a nakloněných plochách jsou rovněž stabilnější než šestinohé mobilní roboty. [4]



Obrázek 9 – Osminohý krácející robot REBUG III

1.2.7 Mikroroboty

Mikrorobotika, někdy také nazývaná nanorobotika, je společně se servisními roboty v současnosti nesporně progresivně se vyvíjející oblastí technické robotiky. Do nynější doby byly vyrobeny mobilní roboty, jejichž rozměry se pohybují nejvíce v desítkách milimetrů. Příkladem je mikrorobot PLIF (Itálie, 1996).

Jedná se o třínohý mobilní robot, jehož dvě nohy jsou poháněny pomocí piezokeramických bimorfních akčních členů a třetí je noha je pasivní. Robot je velmi malý (20 x 20 x 20 mm) a má nízkou hmotnost (30 gramů), ale současně je vzhledem k rozměrům a hmotnosti velmi rychlý (až 20 cm/s). Pohybuje se pomocí velmi malých kroků (několik mikronů), přičemž nohy kmitají vysokou frekvencí (až 800 Hz). Nevýhodou je neurčitost amplitudy a směru pohybu, která podstatně závisí na typu a nerovnostech povrchu. Proto musí být řízení pohybu robotu po trajektorii prováděno v uzavřené smyčce. Pro měření pozice je použit infračervený senzor.

Aplikace těchto robotů lze nalézt v mikromechanice, mikrochirurgii, ostraze a inspekci malých prostor atd.



Obrázek 10 – Mikrorobot PLIF

1.2.8 Plazivé mobilní robotické systémy

Plazivé mobilní roboty se vyznačují specifickou štíhlou podlouhlou konstrukcí. Pro jejich pohyb se vychází z modelů pohybů živočichů a používají se dva typy:

- Had
- Žížala

Počet článků a typ pohonu robotu je volena dle stylu pohybu, předpokládané aplikace a terénu. Počet článků se pohybuje v širokém intervalu od několika jednotek do několika desítek kusů.

Roboty se používají převážně k inspekci a servisní činnosti v potrubích a úzkých prostorech. Plazivé roboty pro tuto činnost se vyznačují vysokým počtem článků.

Další aplikací je průzkum ve velice členitém venkovním prostoru, kde ani pásové ani krácející roboty nemají naději na úspěch. Tyto roboty se vyznačují nižším počtem členů s tuhou konstrukcí a jsou schopny překonávat překážky vyšší, než je polovina délky robotu.

[5]



Obrázek 11- Snake robot number 7

1.2.9 Speciální mobilní robotické systémy

Skákající roboty slouží pro výzkum kinematických a dynamických poměrů jako výchozí bod pro konstrukci složitějších kráčejších robotů (zejména výzkum stability a udržení přímého postoje). Tyto roboty bývají většinou osazeny jednou nohou poháněnou pneumatickým válcem.

Princip pohybu je takový, že robot musí vyskočit dostatečně vysoko, aby se noha mohla překmitnout do nové pozice a robot se posunul o krok (skok) vpřed. Roboty jsou konstrukčně řešeny s hmotným tělem a lehkou kyvnou nohou. Tato koncepce zjednodušuje udržení stability. Příčná stabilita robotu bývá zajištěna vodítkem a robot tak při svém pohybu opisuje kružnici. [6]

1.2.9.1 Šplhající mobilní robotické systémy

Šplhající roboty bývají postaveny na kolové, čtyřnohé nejčastěji však osminohé kráčejší platformě. Osminohá koncepce je pro tyto účely nejvhodnější, protože roboty se mohou pomocí čtyř noh dostatečně stabilně uchytit a zajistit dostatečně pevnou vazbu s povrchem, přičemž ostatní čtyři nohy vykonávají krok vpřed. V případě, že jedna noha ztratí kontakt

s povrchem, musí zbylé tři nohy udržet robota na místě. Šplhající roboty se nejčastěji využívají pro kontrolu, čištění a údržbu výškových budov. [4]



Obrázek 12- Robot čistící skleněnou pyramidu v Louvru

1.2.9.2 Plovoucí robotické systémy

Primárním důvodem pro vznik plovoucích mobilních robotů byla těžba ropy. První roboty byly konstruovány jako teleoperátory s omezeným dosahem, s komunikací užívající vodiče, později optické kabely. Tyto roboty byly určeny pro kontrolu, údržbu a opravy těžních ropných věží. Roboty jsou osazeny manipulátory s potřebnými efekty a silným kamerovým vybavením. Další aplikací podmořských robotů jsou průzkumné činnosti. Jedná se o průzkum povrchu mořského dna, vraků lodí.

Roboty jsou koncipovány jako autonomní s jistým prvkem volného rozhodování. Řízení je však nadále udáváno teleoperátorem. Roboty jsou vybaveny nejrůznějšími snímacími systémy od termovizí a teplotních čidel pro detekci teplých proudů a sopečné činnosti, přes ultrazvukové sonary, TV kamery, atd. Podmořské roboty jsou navrženy tak, aby vydržely pod vodou co nejdéle z jedné zásoby zdrojů energie.

V oblasti vodních mobilních robotů se objevují i roboty plovoucí po hladině. Jedná se například o roboty veslující, které slouží k výzkumným účelům.

Plovoucí roboty mohou sloužit pro ostrahu přístavů nebo pobřeží.



Obrázek 13- Želví robot plavec jménem Madeleine

1.3 Vývojová prostředí

1.3.1 MPLAB C30

MPLAB umožňuje ladění a programování mikrokontrolérových aplikací s dsPIC DSC. Umožňuje nastavení breakpointů nebo zastavení běhu v libovolném okamžiku, nahlížení do registrů mikrokontrolerů a jejich modifikace při zastavení běhu, krokování a to vše nejenom v assembleru ale i v C. MPLAB je sice komerční aplikací, avšak existuje možnost spuštění zdarma pod studentskou licenci.

1.3.2 Pikilab

Linuxová alternativa k programu MPLAB. Pikilab však nabízí práci pouze v assembleru.

1.3.3 Matlab

Nástroje pro Matlab umožňují testovat e-pucka. K programování jsou však nevhodné

II. PRAKTICKÁ ČÁST

2 ZPROVOZNĚNÍ MOBILNÍHO ROBOTY E-PUCK

Pro naprogramování robota e-puck potřebujeme následující:

- Robota e-puck
- Počítač s Bluetooth (například s USB Bluetooth adaptérem)
- Kompilátor (program MPLAB IDE s kompilátorem C30, MPLAB IDE je neplacený program, lze si jej tedy bez problému stáhnout na adrese <http://www.microchip.com>. Kompilátor C30 je sice komerční program, ale pro neziskové účely můžeme využít studentské licence.)
- Bootloader (například *Tiny Bootloader*, jedná se o neplacený program, který je volně ke stažení zde: <http://www.etc.ugal.ro/cchiculita/software/tinybldusage.htm>)

2.1 Rozblikání LED diody (Hello World)

Stejně tak jako mají všechny programovací jazyky, pro vytváření programů na stolní počítače, svůj typický program pro začátečníky, kdy se na obrazovce vypíše, Hello World, tak i první program pro microcontroller PIC oznámí svou funkčnost a to rozblikáním LED diody. Pro zdárné spuštění programu Hello World je zapotřebí splnit tyto tři kroky:

- Vytvoření projektu v programu MPLAB IDE
- Zkompilování kódu
- Naprogramování robota e-puck přes Bluetooth

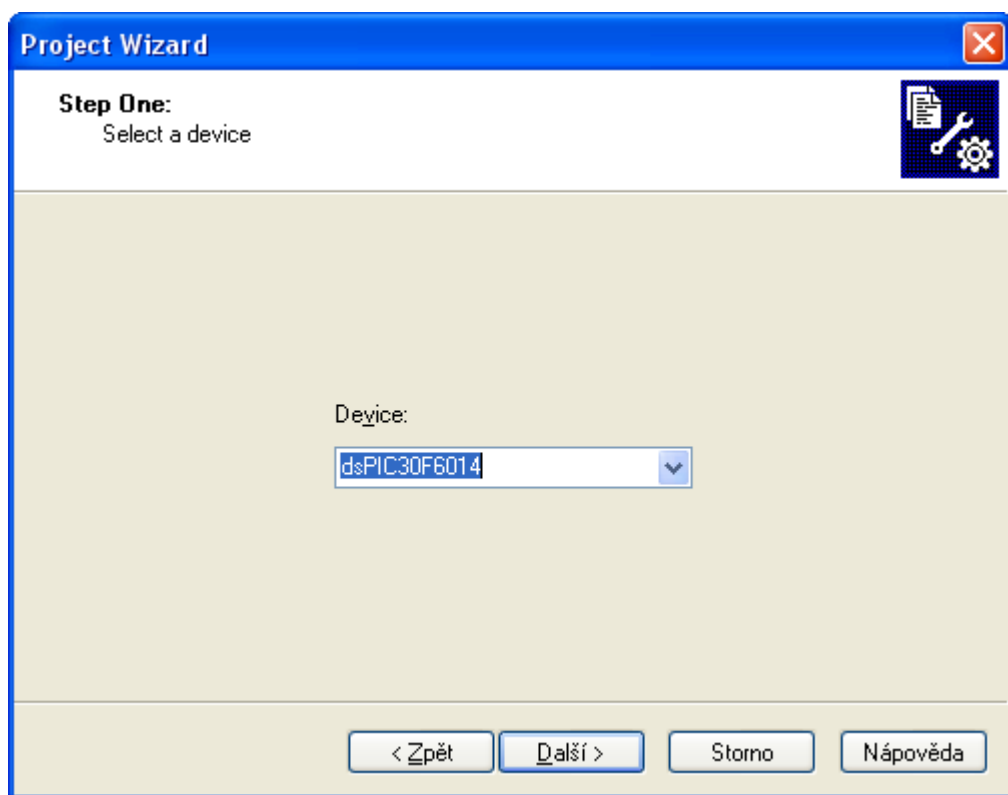
2.1.1 Vytvoření projektu

První krok je vytvoření projektu a pracovního místa zvaného workspace v programu MPLAB IDE. Projekt obsahuje soubory potřebné pro kompilaci aplikace. Obvykle se v jednom workspace nachází pouze jeden projekt.

MPLAB IDE obsahuje funkci Project Wizard která je velkým usnadněním při vytváření nového projektu.

2.1.1.1 Výběr zařízení

1. Spustit program MPLAB IDE
2. Zavřít všechny otevřené projekty (File > Close Workspace)
3. Spustit Project Wizard (Project menu > Project Wizard)
4. Odkliknout uvítací okno (Next>)
5. Zobrazí se okno s názvem *Step One* ve kterém je třeba vybrat označení mikrokontroleru (dsPIC30F6014) a pokračovat na další obrazovku (Next>)



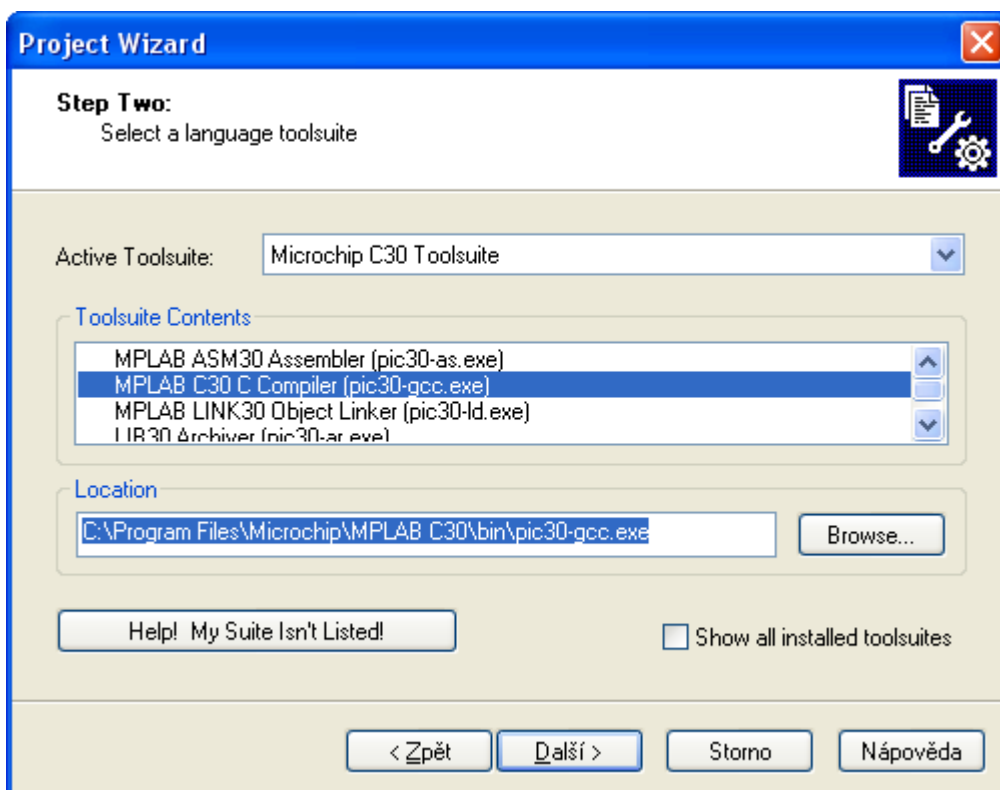
Obrázek 14- Project Wizard, krok 1, výběr zařízení

2.1.1.2 Výběr pracovního balíčku

1. Z menu *Aktive toolsuite* vybrat možnost *Microchip C30 Toolsuite*. Tento nástrojový balíček obsahuje nástroje, které budou použity
2. V bloku *Toolsuite Contents* vybrat *MPLAB ASM30 Assembler (pic30-as.exe)*
3. V bloku *Location* kliknout na *Browse* a nastavit:

C:\Program Files\Microchip\MPLAB C30\bin\pic30-as.exe

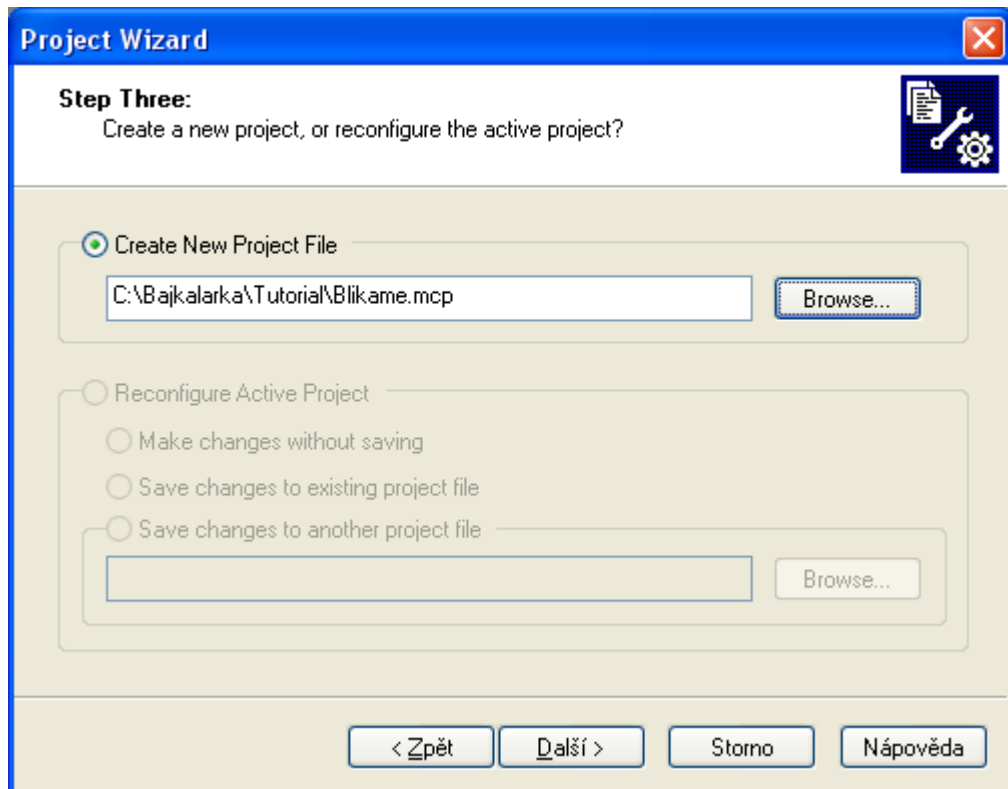
4. V bloku *Toolsuite Contents* vybrat MPLAB C30 Compiler (pic30-gcc.exe)
5. V bloku *Location* kliknout na *Browse* a nastavit:
C:\Program Files\Microchip\MPLAB C30\bin\pic30-gcc.exe
6. V bloku *Toolsuite Contents* vybrat MPLAB LINK30 Linker (pic30-ld.exe)
7. V bloku *Location* kliknout na *Browse* a nastavit:
C:\Program Files\Microchip\MPLAB C30\bin\pic30-ld.exe
8. V bloku *Toolsuite Contents* vybrat LIB30 Archiver (pic30-ar.exe)
9. V bloku *Location* kliknout na *Browse* a nastavit:
C:\Program Files\Microchip\MPLAB C30\bin\pic30-ar.exe
10. Pokračovat na další obrazovku (Next>)



Obrázek 15- Project Wizard, krok 2, výběr pracovního balíčku

2.1.1.3 Jméno projektu

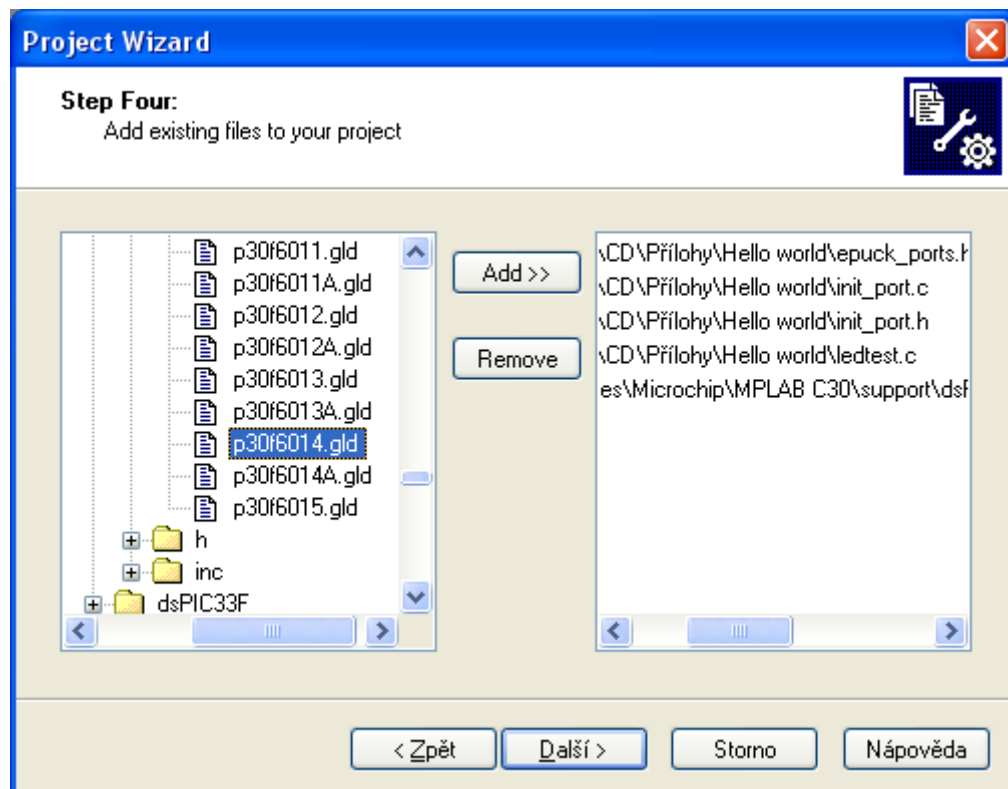
1. V bloku *Create New Project File* kliknout na *Browse* a zvolit cestu kam má Project Wizard vytvořit nový projekt.
2. Pokračovat na další obrazovku (Next>)



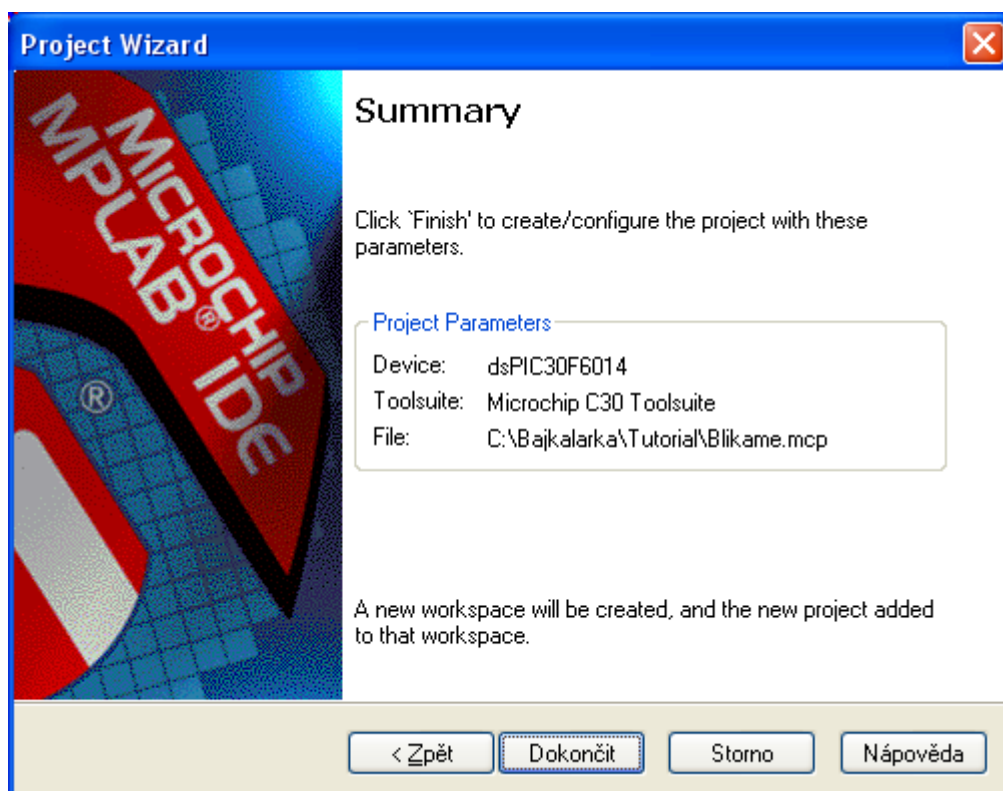
Obrázek 16- Project Wizard, krok 3, výběr workspace, volba jména projektu

2.1.1.4 Přidat do projektu existující soubory

1. Vyhledání souborů z přílohy této bakalářské práce (například `D:\přílohy\Hello world`) a označení souboru `ledtest.c`
2. Stisknutím tlačítka `Add>>` se vybraný soubor připojí k projektu.
3. To samé se opakuje u souborů `init_port.c`, `init_port.h` a `epuck-ports.h`.
4. V umístění `C:\ProgramFiles\Microchip\MPLAB C30\support\dsPIC30F\gld` vybereme soubor `p30f6014.gld`
5. Po přidání všech souborů se pokračuje na poslední okno `Next>`
6. Poslední okno slouží k poslední kontrole zadaných údajů před vytvořením projektu tlačítkem `Finish`



Obrázek 17- Project Wizard, krok 4, přidání existujících souborů do projektu



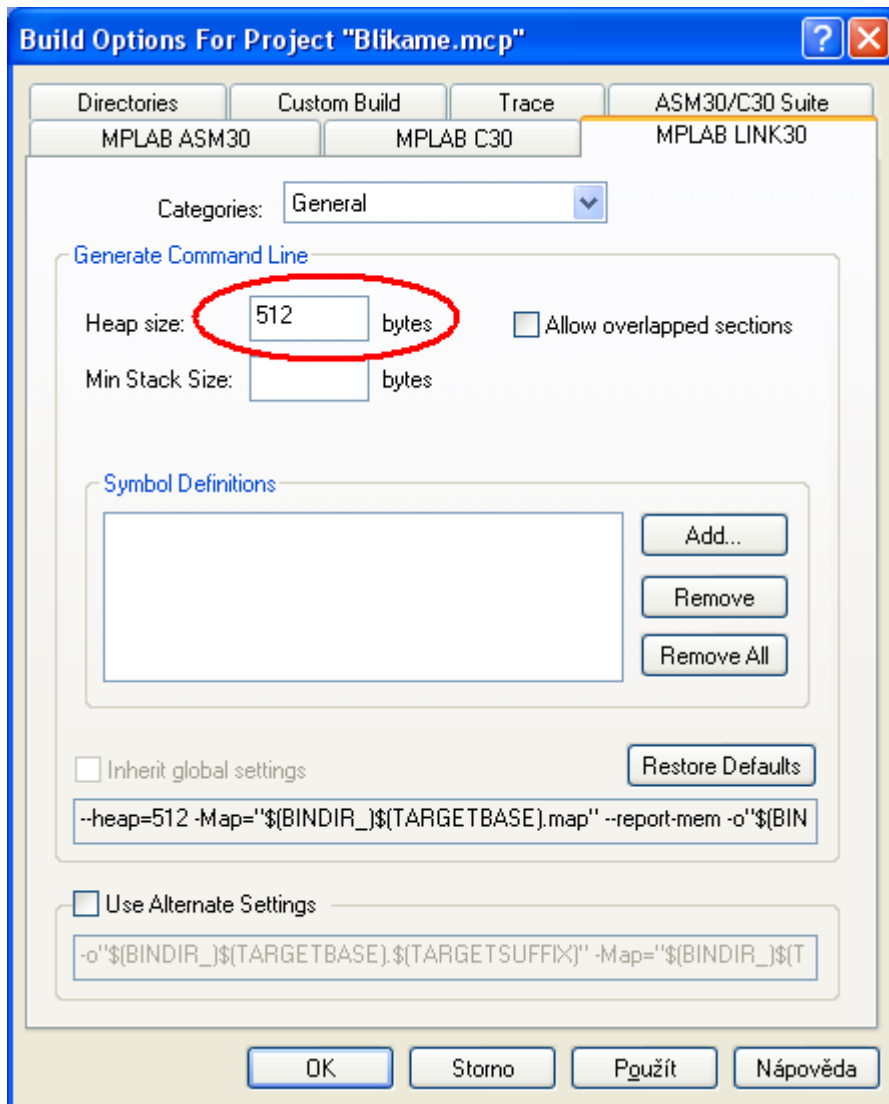
Obrázek 18- Project Wizard, krok 5, kontrola zadaných údajů

2.1.2 Zkompilování projektu

Pro zdárnou kompilaci je potřeba upravit nastavení několika položek.

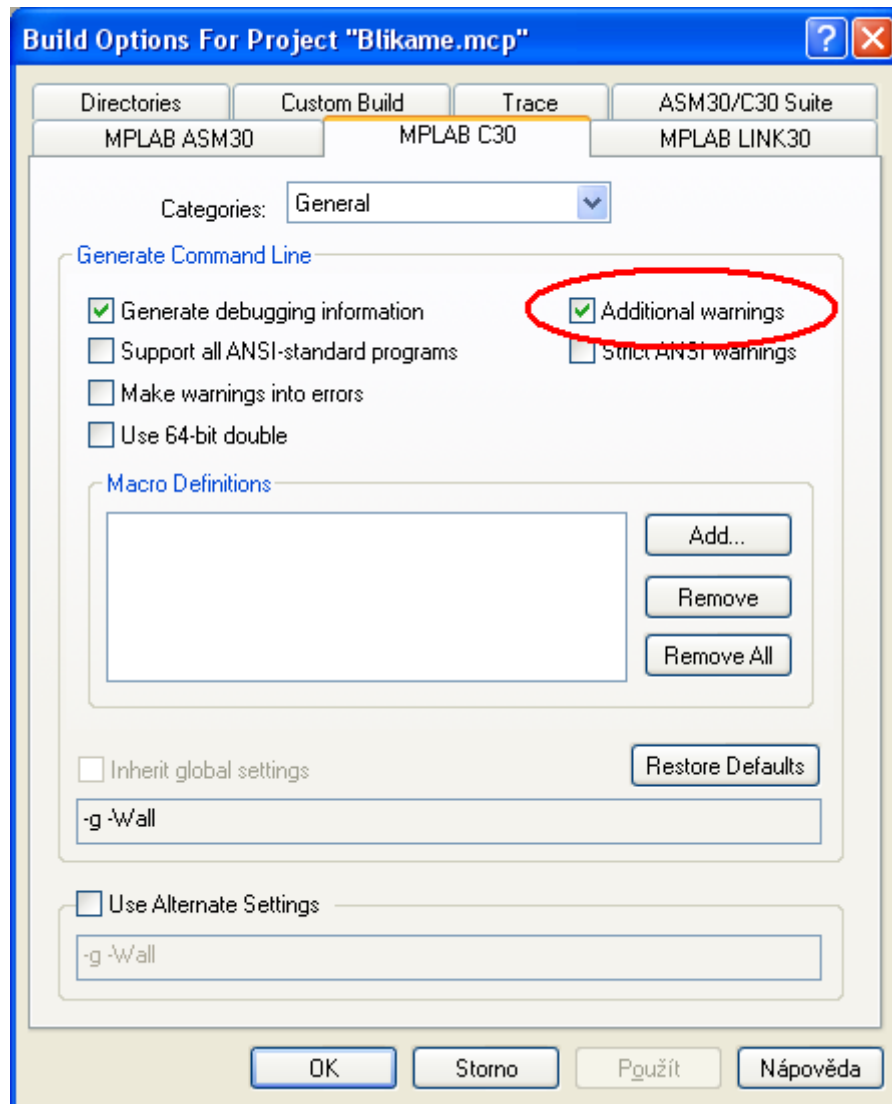
2.1.2.1 Nastavení kompilačních parametrů

1. V menu pro kompilaci `Project > Build Options > Project` je třeba nastavit velikost hlavičky na 512 bytů



Obrázek 19- Nastavení velikosti hlavičky kompilátoru

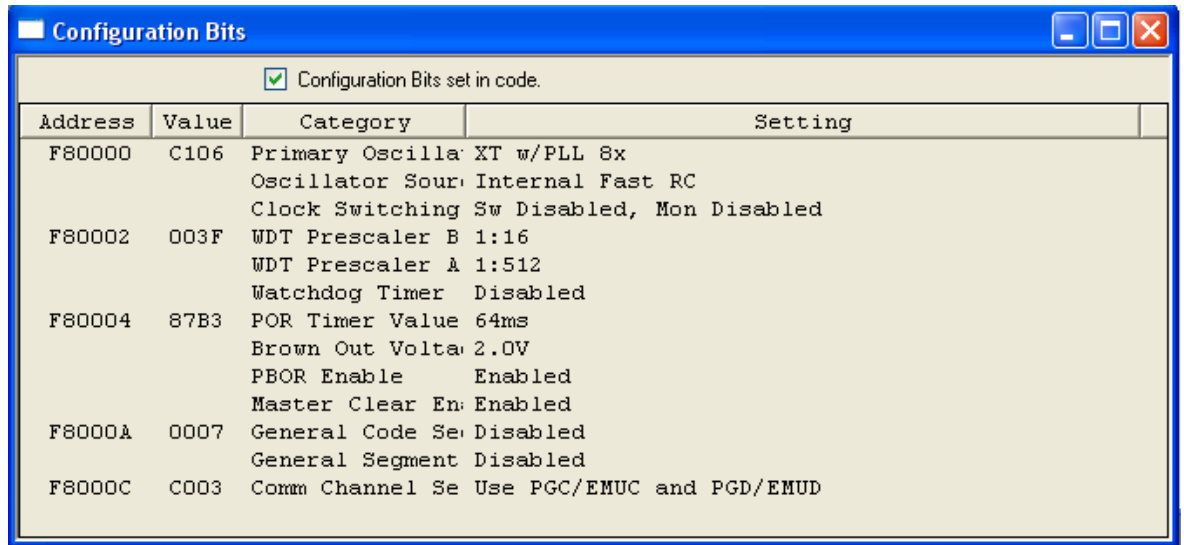
2. Pro podrobnější hlášení je potřeba v okně pro kompilaci `Project > Build Options > Project` zatrhnout `Additional warnings`



Obrázek 20- Nastavení podrobnějšího výpisu hlášení

3. Nastavení oscilátoru a vypnutí Watchdoga provedeme v okně Configure > Configuration Bits

- Oscilátor: XT w/PLL 8x
- Watchdog Timer: Disabled



Configuration Bits

Configuration Bits set in code.

Address	Value	Category	Setting
F80000	C106	Primary Oscilla	XT w/PLL 8x Oscillator Sour: Internal Fast RC Clock Switching Sw Disabled, Mon Disabled
F80002	003F	WDT Prescaler B	1:16 WDT Prescaler A 1:512 Watchdog Timer Disabled
F80004	87B3	POR Timer Value	64ms Brown Out Volta: 2.0V PBOR Enable Enabled Master Clear En: Enabled
F8000A	0007	General Code Se	Disabled General Segment Disabled
F8000C	C003	Comm Channel Se	Use PGC/EMUC and PGD/EMUD

Obrázek 20- Nastavení oscilátoru a vypnutí Watchdoga

2.1.2.2 *Samotná kompilace*

1. Kompilace se spustí Project > Build All nebo klávesovou zkratkou Ctrl + F10
2. Probíhá kompilace
3. Pokud se na konci kompilace vypíše BUILD SUCCEEDED tak proběhla kompilace v pořádku a program je připraven k zavedení do robota e-puck

```

Output
Build Version Control Find in Files
ledtest_int.c:37: warning: implicit declaration of function 'InitPort'
Executing: "C:\Program Files\Microchip\MPLAB C30\bin\pic30-gcc.exe" -mcpu=30F6014 "init_port.o" "ledtest_int

Program Memory Usage
-----
section                address      length (PC units)  length (bytes) (dec)
-----
.reset                 0            0x4                0x6 (6)
.ivt                   0x4          0x7c               0xba (186)
.aivt                  0x84         0x7c               0xba (186)
.text                  0x100        0x88               0xcc (204)
.text                  0x188        0x1b6              0x291 (657)
.dinit                 0x33e        0x2                0x3 (3)
.isr                   0x340        0x2                0x3 (3)

Total program memory used (bytes):          0x4dd (1245) <1%

Data Memory Usage
-----
section                address      alignment gaps    total length (dec)
-----
Total data memory used (bytes):              0 (0)

Dynamic Memory Usage
-----
region                address      maximum length (dec)
-----
heap                   0x800        0x200 (512)
stack                  0xa00        0x1da0 (7584)

Maximum dynamic memory (bytes):              0x1fa0 (8096)

Executing: "C:\Program Files\Microchip\MPLAB C30\bin\pic30-bin2hex.exe" "C:\Bajkalarka\Tutorial\e-puck\tut
Loaded C:\Bajkalarka\Tutorial\e-puck\tutorial\tutorial B\Treti.cof.

Debug build of project 'C:\Bajkalarka\Tutorial\e-puck\tutorial\tutorial B\Treti.mcp' succeeded.
Preprocessor symbol '__DEBUG' is defined.
Tue May 19 15:31:35 2009

BUILD SUCCEEDED

```

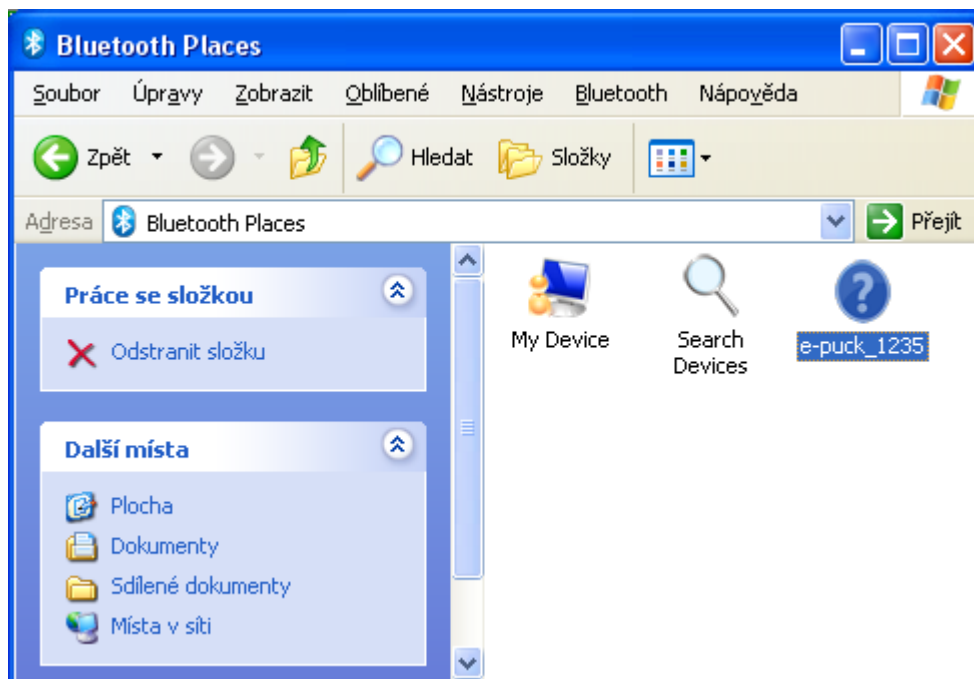
Obrázek 21- Kompilace proběhla úspěšně

2.1.3 Naprogramování robota e-puck

Aby bylo možné robota naprogramovat, je zapotřebí nejprve propojit e-pucka s počítačem. Toto propojení se provádí bezdrátově pomocí Bluetooth

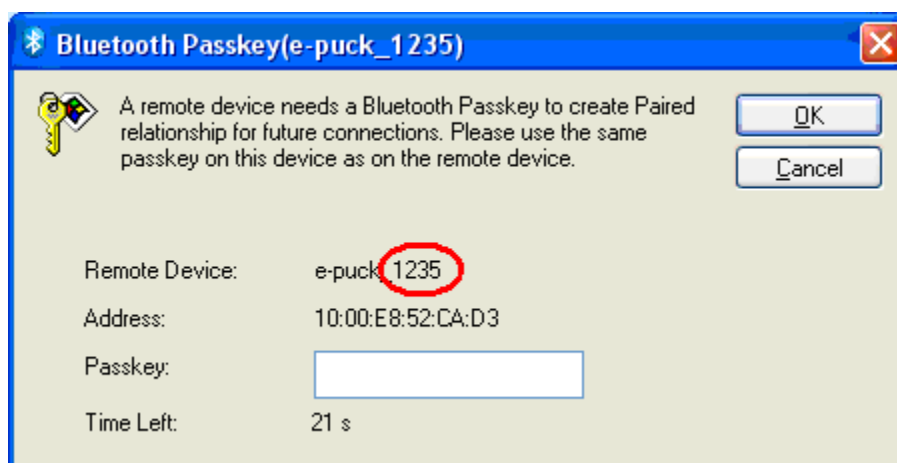
2.1.3.1 Spárování robota e-puck přes Bluetooth

1. Zapnutí robota. Svítí zelená LED dioda.
2. Zapnutí Bluetooth na počítači
3. Vyhledání další Bluetooth zařízení v okolí počítače (Search device)

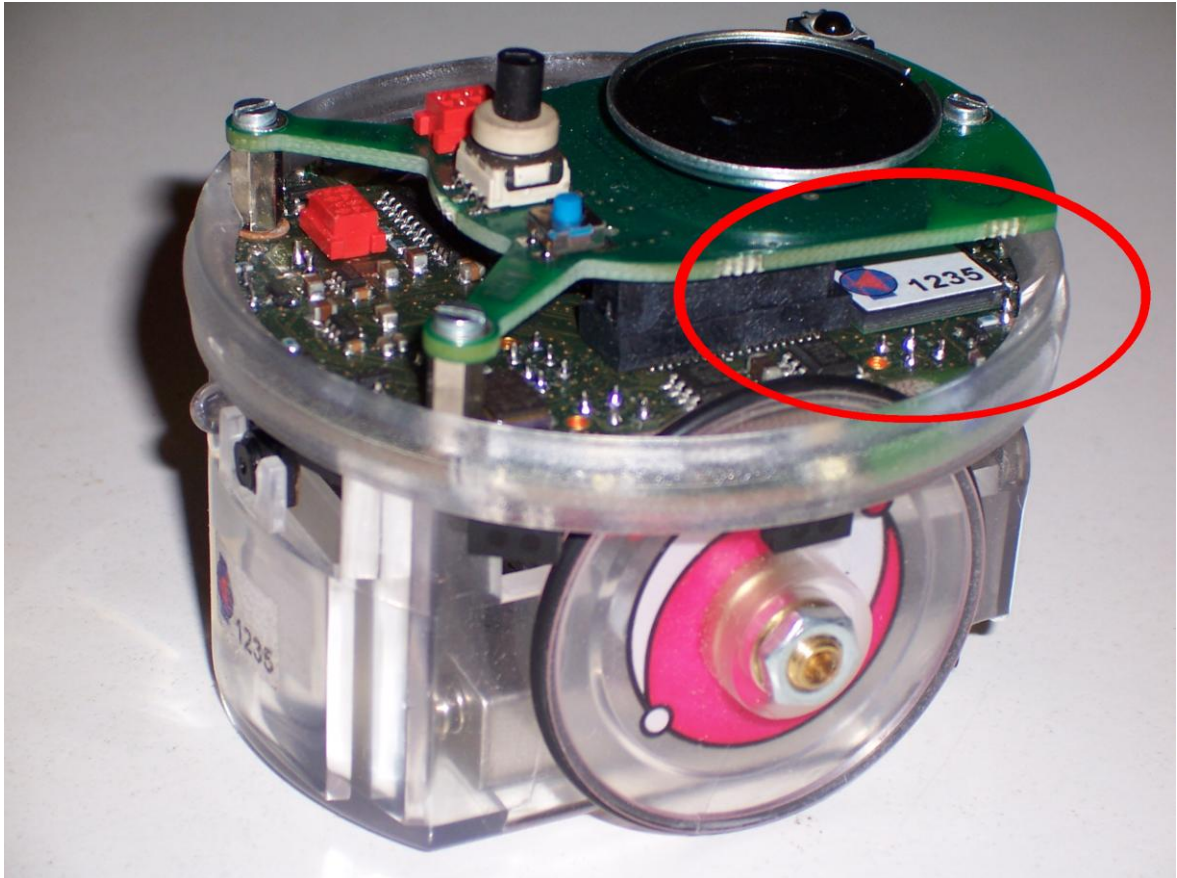


Obrázek 22- Vyhledání dalších Bluetooth zařízení

4. Každý e-puck má své vlastní identifikační číslo, lze tedy s jistotou říci, zda počítač vyhledal požadovaného robota. Vyhledaný e-puck se ohlásí jako e-puck_XXXX. Kdy XXXX jsou právě ono identifikační číslo, toto číslo nalezneme napsané také na samotném robotovi a to vždy na štítku přilepeném na základní desce pod reproduktorem. V našem případě se jedná o číslo 1235. Toto číslo je také použito jako heslo.

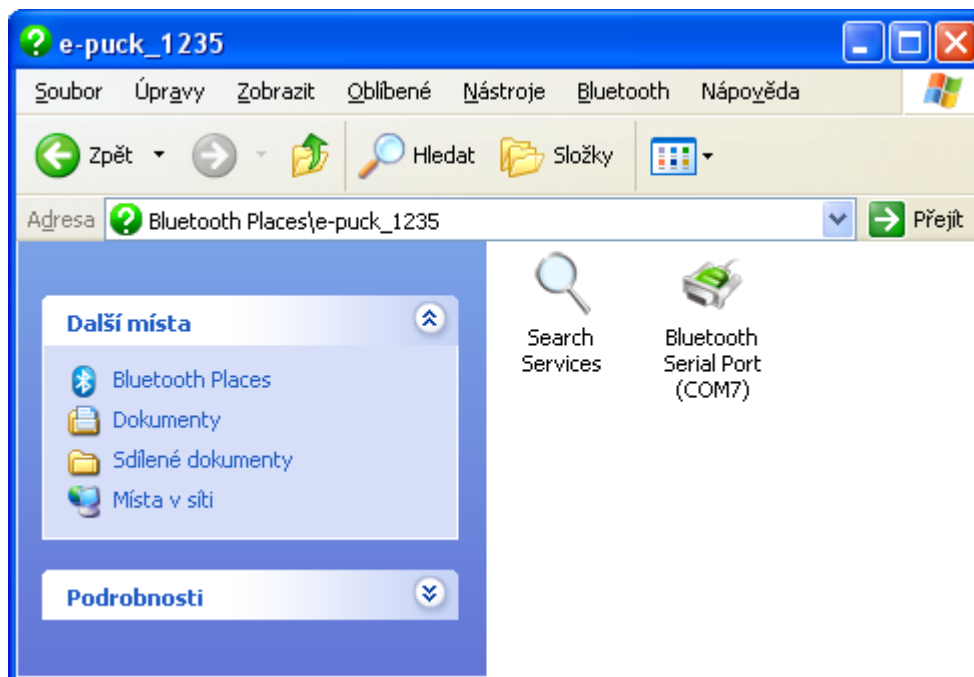


Obrázek 23- Zadání hesla pro spojení



Obrázek 24- Robot e-puck a jeho identifikační číslo

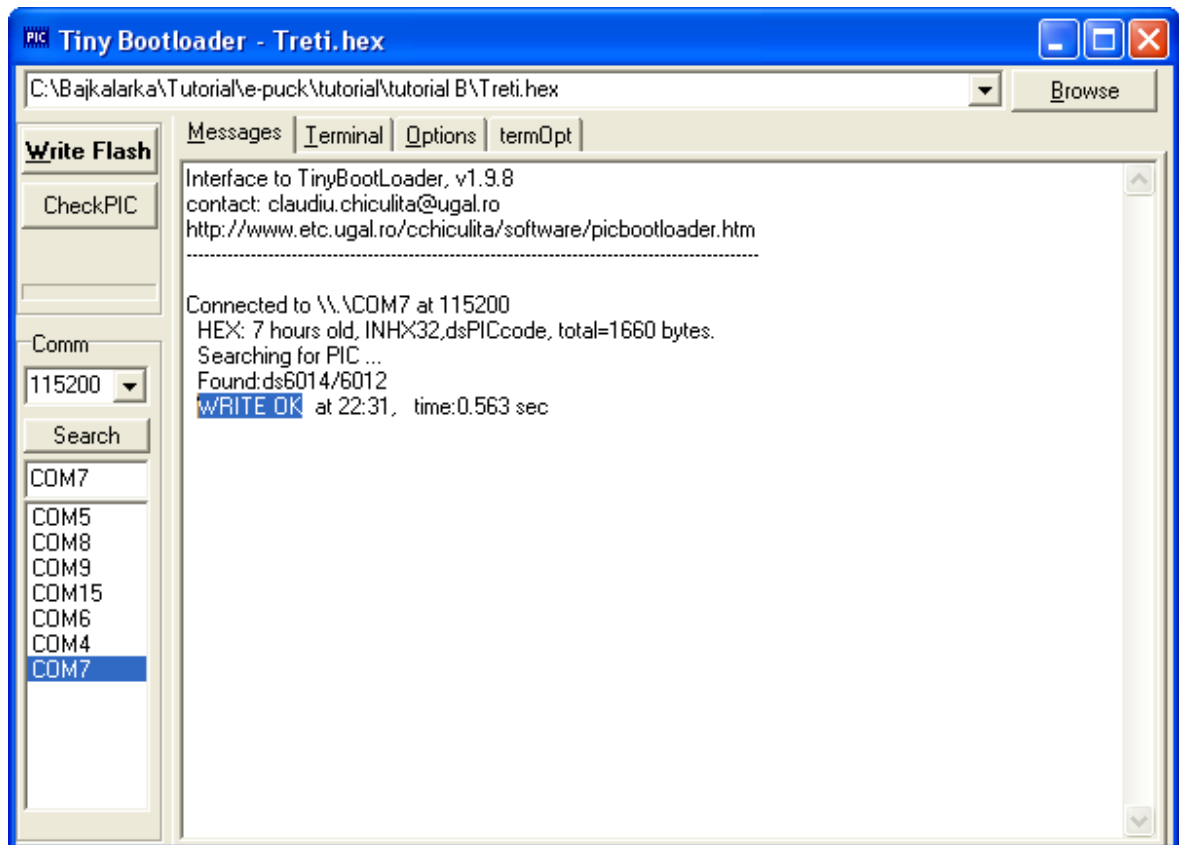
5. Po přijetí hesla se ve složce Bluetooth Places\ e-puck_XXXX nachází virtuální sériový port Bluetooth Seriál Port (COM 7). Dvojklikem na tuto ikonu se připojí robot k počítači. Pokud vše proběhlo bez problémů, ikonka sériového portu změní barvu z modré na zelenou.



Obrázek 23- Spojení s robotem je navázáno

2.1.3.2 Flashnutí robota e-puck

1. Zapnout program Tiny Bootloader
2. Po kliknutí na tlačítko Browse vyhledat soubor končící na příponu .hex
3. V levém dolním sloupci je potřeba vybrat port COM, na který je připojen robot. V našem případě COM7
4. Po stisknutí tlačítka Write Flash je nutné do pěti sekund stisknout modré tlačítko reset na robotovy a tím potvrdit požadavek na zavedení nového programu. Pokud nebude tlačítko včas zmáčknuto e-puck se sám odpojí od počítače. A bude nutné znovu navázat spojení.
5. Pokud vše proběhlo úspěšně, vypsal program Tiny Bootloader: WRITE OK a robot e-puck se rozblíkal.



Obrázek 24- Program Tiny Bootloader

2.2 Knihovny

Pro Mobilního robota e-puck jsou ke stažení také vytvořené knihovny, které velice usnadňují programování aplikací. Zde je stručný popis pár základních knihoven.

2.2.1 e_init_ports

Knihovna obsahuje funkci, která ziniculuje porty. Není proto nutné vypisovat inicializaci všech portů v každém programu znovu.

2.2.2 e_led

Knihovna obsahující funkce pro správu LED diod.

2.2.3 e_motors

Knihovna obsahující funkce pro správu motorů. Stačí pouze zavolat příslušnou funkci, zadat parametr rychlosti a motor začne otáčet kolem požadovanou rychlostí. Není třeba se starat o každou fázi pootočení krokovým motorkem.

2.2.4 e_sound

Knihovna zabývající se zprávou zvuku. Obsahuje kodeky a předvytvořené zvuky. Stačí tedy pouze zavolat funkci s číslem zvuku, jeho požadovanou délkou a knihovna sama se již postará, aby byl zvuk přehrán. Výpis některých zvuků z knihovny a jejich název:

- [0, 2112]: "haa"
- [2116, 1760]: "spaah"
- [3878, 3412]: "ouah"
- [7294, 3732]: "yaouh"
- [11028, 8016]: "wouaaaaaaaah"

2.2.5 P30f6014A.gld

Tato knihovna je nejdůležitější pro samotnou kompilaci aplikace, neboť obsahuje informace o mikrokontroleru.

ZÁVĚR

Cílem diplomové práce bylo poskytnout přehledný návod pro programování mobilního robota e-puck v programovém prostředí MPLAB s kompilátorem C30. Teoretická část práce se zaměřila na popis senzorických a lokomočních ústrojí mobilních servisních robotů a jejich využití v praxi. Dále pak na popis programových prostředí, ve kterých lze mobilního robota e-puck programovat. Z těchto programovacích prostředí byl vybrán program MPLAB pro svou pohodlnost, přehlednost a hlavně dostupnost pod studentskou licenci. Kompilátor C30 byl vybrán, protože programy v něm stačí psát v jazyce C++ a není třeba psát program v assembleru. Toto neznamená, že by znalost programovacího jazyku nebyla potřebná, některé části knihovny jsou napsány pouze v jazyku assembleru a pro zjištění jejich funkcí je potřeba tento programovací jazyk ovládat. Jedinou nevýhodou programu MPLAB spuštěným pod studentskou licenci je, že zkompileovaný kód není optimalizován, zpoždění získané během neoptimalizované aplikace je však minimální. Stručný návod, který zahajuje praktickou část této práce, čtenáře provede krok za krokem založením projektu v programu MPLAB, přiložením všech potřebných souborů, jako jsou například knihovny, nastavení kompilace projektu a jeho samotnou kompilací do zdrojového kódu, jazyka kterému rozumí mikrokontroler dsPIC 30F6014A hlavní čip mobilního robota e-puck, propojení mobilního robota e-puck přes Bluetooth a vytvoření virtuálního sériového portu, nahrání kompilací získaného programu přes virtuální sériový port do mobilního servisního robota e-puck pomocí programu Tiny Bootloader. Dále se praktická část zabývá popisem funkcí vybraných knihoven používaných při tvorbě aplikací pro mobilního robota e-puck.

ZÁVĚR V ANGLIČTINĚ

The purpose of bachelor thesis was to create transparent instruction for programming mobile robot e-puck in MPLAB program with C30 compiler. Theoretical part was focused on description of sensorial and locomotive mobile mechanism of service robots and its practical use. Below was mentioned description of programs in which can be mobile robot e-puck programmed. From these was chosen MPLAB program, which is comfortable and transparent and it has free licence for students. Compiler C30 was chosen, because programs can be written in C++ language. This doesn't mean that knowledge of assembler language isn't needed. Some libraries are written only in this language, so this language is also needed. Only disadvantage of MPLAB program run under student licence is, that compiled code isn't optimized, but delay of unoptimized application is minimal. Brief instructions, of writing project in MPLAB, which are included in practical part, takes reader step by step. Also inclosing of needed files, like libraries, settings of project compilation and its implementation to source code and language which is known by microcontroler dsPIC 30F6014A main chip of mobile robot e-puck, connection of robot through Bluetooth and creation of virtual serial port, uploading compilation of program through virtual serial port to mobile service robot e-puck with help of Tiny Bootloader program. Practical part also contains description of functions of selected libraries used for creating applications for mobile robot e-puck.

SEZNAM POUŽITÉ LITERATURY

- [1] KÁRNÍK, L. KONEČNÝ, Z.: Special of translation modules of mobile service robots. In: Robtep 2001, Prešov, SF TU Košice 2001, s. 63-68. ISBN 80-7099-749-4.
- [2] KÁRNÍK, L. – BUZEK, V.: Lokomoční ústrojí servisních robotů pro indoor aplikace. In: Setkání ústavů a kateder oboru výrobní stroje a robotika. FS ČVUT, Praha, 2003, s. II/34–II/36. ISBN 80-01-02815-1.
- [3] SKAŘUPA, J. – MOSTÝN, V.: Metody a prostředky návrhu průmyslových a servisních robotů. 1. vydání. Vienala, Edícia vedeckej a odbornej literatúry, Košice. Strojnícka fakulta TU v Košiciach, 2002, 190 s. ISBN 80-88922-55-0.
- [4] Konstrukce krácejících mobilních robotů. MM : Průmyslové spektrum. 2001, č. 1, s. 65
- [5] HEKRDLA, Michal. Řízení pásového robota s využitím motorů MAXO. Praha, 2007. 61 s. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE. Diplomová práce.
- [6] PETR, Ondřej. Řídicí systém mobilního dvoustopého robota. Zlín, 2006. 79 s. UTB ve Zlíně, Fakulta aplikované informatiky. Vedoucí diplomové práce Mgr. Ing. Milan Kvasnica, CSc.
- [7] EVERETT, H.,R.: Sensors for Mobile Robots, Theory and Applications. 1.vyd. A.K.Peters, Ltd. ISBN 1-568811-048-2.

SEZNAM OBRÁZKŮ

Obrázek 1-Diferenčně řízený robot.....	15
Obrázek 2-Tříkolový robot řízený Ackermanovým způsobem.....	16
Obrázek 3-Synchronně řízený robot.....	16
Obrázek 4-Mobilní sonda Sojourner (NASA).....	17
Obrázek 5-Čtyřpásový mobilní robot.....	19
Obrázek 6-Mobilní kráčejíci robot Honda Asimo P3.....	20
Obrázek 7-Robot savec jménem BigDog	21
Obrázek 8-Hmyzí robot CWRU III.....	22
Obrázek 9-Osminohý kráčejíci robot REBUG III.....	23
Obrázek 10-Mikrorobot PLIF.....	24
Obrázek 11- Snake robot number 7.....	25
Obrázek 12- Robot čistící skleněnou pyramidu v Louvru.....	26
Obrázek 13- Želví robot plavec jménem Madeleine	27
Obrázek 14- Project Wizard, krok 1, výběr zařízení.....	30
Obrázek 15- Project Wizard, krok 2, výběr pracovního balíčku.....	31
Obrázek 16- Project Wizard, krok 3, výběr workspace, volba jména projektu.....	32
Obrázek 17- Project Wizard, krok 4, přidání existujících souborů do projektu.....	33
Obrázek 18- Project Wizard, krok 5, kontrola zadaných údajů.....	33
Obrázek 19- Nastavení velikosti hlavičky kompilátoru.....	34
Obrázek 20- Nastavení podrobnějšího výpisu hlášení.....	35
Obrázek 20- Nastavení oscilátoru a vypnutí Watchdoga.....	36
Obrázek 21- Kompilace proběhla úspěšně.....	37
Obrázek 22- Vyhledání dalších Bluetooth zařízení.....	38
Obrázek 23- Zadání hesla pro spojení.....	39

Obrázek 24- Robot e-puck a jeho identifikační číslo.....	39
Obrázek 25- Spojení s robotem je navázáno.....	40
Obrázek 26- Program Tiny Bootloader.....	41

SEZNAM PŘÍLOH

Příloha P I: Hlavní funkce

Příloha P II: Funkce pro detekci potencionální kolize

Příloha P III: Funkce pro detekci kolize a pádu

PŘÍLOHA P I: HLAVNÍ FUNKCE

```
void run _shocker(void)
{
    e_init_ad_scan(MICRO_ONLY);
    init_micro();
    e_init_port();
    e_init_motors();
    e_init_ad_scan(ALL_ADC);
    e_init_sound();
    e_calibrate_ir();
    e_activate_agenda(flow_led, 900);
    e_activate_agenda(shock, 650);
    e_start_agendas_processing();

    while(1) {
        e_start_agendas_processing();
        run_accelerometer();
    }
}
```

PŘÍLOHA P II: FUNKCE PRO DETEKCI POTENCIONÁLNÍ KOLIZE

```
void shock(void)
{
    for (m = 0; m < 2; m++)
    {
        potential[m] = 0;
        for (s = 0; s < 8; s++)
            potential[m] +=
(matrix_prox[m][s]*e_get_calibrated_prox(s)); // get values from
proximity sensors
        speed[m] = (potential[m]/PROXSCALING_SHOCK + BASICSPEED);
    }

    if((speed[1] < 50 && speed[1] > -50)
        && (speed[0] < 50 && speed[0] > -50)) {
        speed[1] = speed[1] * 20;
        speed[0] = speed[0] * 20;
    }

    if (speed[1] > 1000)
        speed[1] = 1000;
    else if (speed[1] < -1000 )
        speed[1] = -1000;

    if (speed[0] > 1000)
        speed[0] = 1000;
    else if (speed[0] < -1000 )
        speed[0] = -1000;

    e_set_speed_left(speed[1]);
    e_set_speed_right(speed[0]);
}
```

PŘÍLOHA P III: FUNKCE PRO DETEKCI KOLIZE A PÁDU

```
void run_accelerometer() {
    e_init_port();
    e_init_ad_scan(ALL_ADC);
    e_init_sound();
    e_init_motors();
    e_start_agendas_processing();
    e_set_led(8, 1);
    e_acc_calibr();
    e_set_led(8, 0);
    e_set_speed_left(100);
    e_set_speed_right(100);
    state=STATE_NORMAL;
    amplavg=1000;
    while (1) {

        accx = e_get_acc(0);
        accy = e_get_acc(1);
        accz = e_get_acc(2) + 744; //744 is 1g

        if ((accz<0) && (accz>-600)) {accz=0;}

        ampl=((long) (accx)*(long) (accx))+((long) (accy)*(long) (accy))+((long)
) (accz)*(long) (accz));
        amplavg=(amplavg>>2)+ampl;

        if (! e_dci_unavailable) {
            if (state!=STATE_NORMAL) {
                state=STATE_NORMAL;
                e_set_led(8, 0);
                e_set_body_led(0);
            }
        }

        if (amplavg<5000) {
            if (state!=STATE_FREEFALL) {
                state=STATE_FREEFALL;
                e_stop_flag=1;
                while (e_dci_unavailable);
                sprintf(buffer, "Free fall: %ld, (%d, %d, %d) ->
(%ld)\r\n", amplavg, accx, accy, accz, ampl);
                e_send_uart1_char(buffer, strlen(buffer));
            }
        }
    }
}
```

```
        e_play_sound(11028, 8016);
        e_set_body_led(1);
        e_set_led(8, 0);
    }
} else if (amplavg>4000000) {
    if (state!=STATE_SHOCK) {
        state=STATE_SHOCK;
        e_stop_flag=1;
        while (e_dci_unavailable);
        sprintf(buffer, "Shock: %ld, (%d, %d, %d) ->
(%ld)\r\n", amplavg, accx, accy, accz, ampl);
        e_send_uart1_char(buffer, strlen(buffer));
        soundsel=(accx & 3);
        if (soundsel==0) {
            e_play_sound(0, 2112);
        } else if (soundsel==1) {
            e_play_sound(2116, 1760);
        } else {
            e_play_sound(3878, 3412);
        }
    }
    e_set_body_led(1);

    angle=atan2(accy, accx);
    lednum=floor(atan2(accy, accx)/PI*4+PI/2+PI/8);
    while (lednum>8) {lednum=lednum-8;}
    while (lednum<0) {lednum=lednum+8;}
    sprintf(buffer, "(x=%d, y=%d) -> angle=%f,
led=%d\r\n", accx, accy, angle, lednum);
    e_send_uart1_char(buffer, strlen(buffer));
    e_set_led(lednum, 1);
}
}
}
}
```