

Robotchallenge 2010 - víťazný robot v kategórii puck collect

Škultéty Jozef · Elektrotechnika, Študentské práce

27.07.2011



RobotChallenge je súťaž robotov, ktorá sa každoročne koná vo Viedni. Ako tím sme sa tejto súťaže zúčastnili v kategórii Puck Collect v roku 2010. Cieľom článku je opísať, ako sme postupovali pri hardvérovej a softvérovej realizácii mobilného robota pre túto súťažnú kategóriu.

1. Úvod

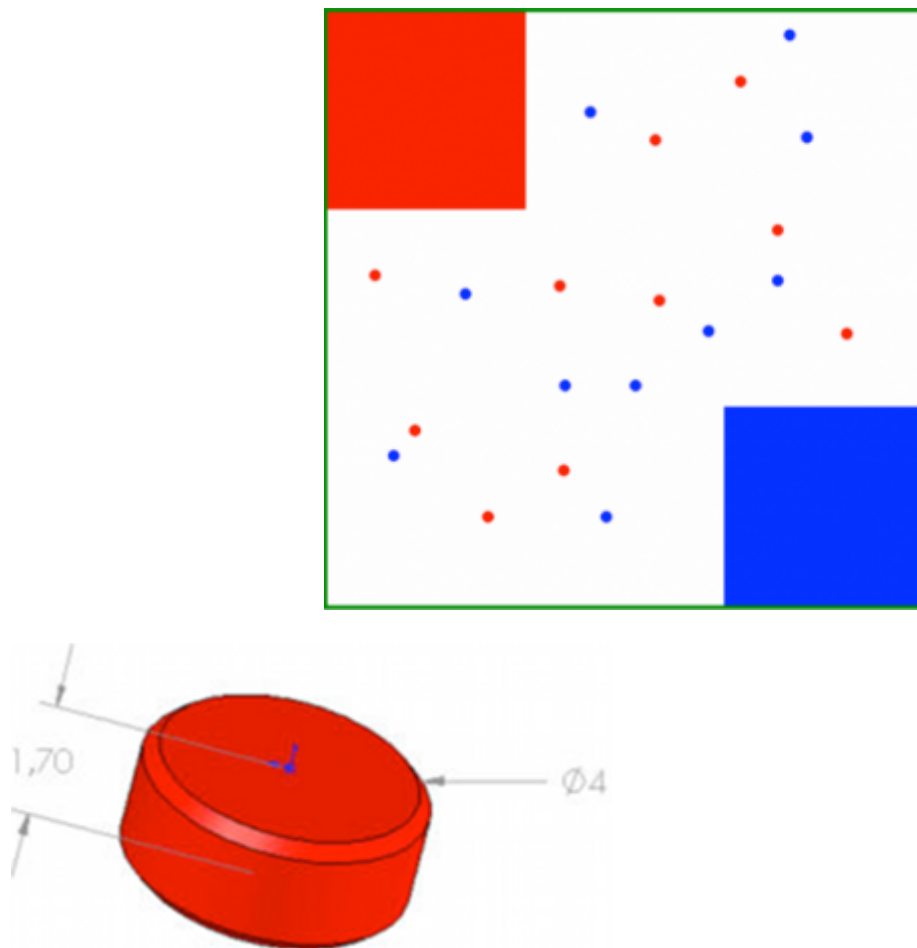
Naším cieľom bolo, zúčastniť sa v roku 2010 ako súťažný tím s robotom súťaže RobotChallenge vo Viedni v kategórii Puck Collect. RobotChallenge je európsky šampionát pre skonštruované autonómne mobilné roboty. Táto súťaž sa každoročne koná vo Viedni. V roku 2010 bolo registrovaných viac ako 600 robotov z 23 krajín. V kategórii Puck Collect bolo registrovaných 21 tímov, pričom na súťaž nastúpilo 13 tímov. V článku je opísaná analýza problému spolu s hardvérovou a softvérovou realizáciou mobilného robota.

2. Pravidlá Puck Collect [11]

Vždy súťažia dvaja roboti. Na hracej ploche sú náhodne rozmiestnené malé farebné puky. Cieľom súťaže je pozbierať všetky puky vlastnej farby a priniesť ich do svojej domovskej základne. Rozmery robotov pre túto súťažnú kategóriu nesmú presiahnuť rozmery 50 cm × 50 cm. Výška robota je neobmedzená.

2.1 Hracia plocha

Hracia plocha (ľavá časť obr. 1) je o rozmeroch 280 cm × 280 cm. Plocha je zároveň ohraničená farebnou 10 cm vysokou doskou. Na hracej ploche sa nachádzajú 2 farebné plochy označujúce domovské základne robotov (červená a modrá). Tieto plochy sú o rozmeroch 70 cm × 70 cm a sú umiestnené v protilaňých rohoch. Zvyšná časť hracej plochy je biela. Na hracej ploche je náhodne rozmiestnených 10 pukov každej farby. Puky sú drevené o veľkosti čajovej sviečky (pravá časť obr. 1).



Obr. 1: Hracia plocha a červený puk [11]

2.2 Hra

Na začiatku hry je každému robotovi priradená farba (buď modrá alebo červená) a každý z dvojice robotov je umiestnený do svojej domovskej základne. Po oznámení začiatku hry rozhodcami môžu byť spustené roboty. Cieľom súťaže je pozbierať všetky puky priradenej farby a doniesť ich do svojej domovskej základne. Ak robot prinesie do svojej domovskej základne všetky puky vlastnej farby, tak tento robot vyhral hru. V prípade, ak sa ani jednému z robotov nepodariť priniesť všetky puky svojej farby do svojej domovskej základne, hra bude po 3 minútach ukončená. V tomto prípade vyhrá ten robot, ktorý má viac svojich pukov vo svojej domovskej zóne.

Puk sa počíta za pozbieraný, ak sa puk nachádza v nehybnom stave a celým objemom nad priestorom v domovskej zóne rovnakej farby. V prípade, ak robot dopraví svoj puk do súperovej domovskej základne, bude tento puk vyradený rozhodcom z hry. V prípade, ak robot dopraví súperov puk do svojej domovskej základne, bude takýto puk presunutý do súperovej domovskej základne.

2.3 Bodové ohodnotenie

Robot, ktorý zvíťazí v hre získá 3 body. V prípade remízy robotov získá každý z robotov 1 bod. V prípade, ak sa hrá vyradovacím systémom nie je možná remíza. Ak oba roboty majú zhromaždené rovnaké množstvo pukov, tak zvíťazí ten robot, ktorý ako prvý doniesol puk do svojej domovskej zóny. Ak ani jeden z robotov nedokáže zhromaždiť aspoň jeden puk počas hry, tak nepostúpi ani jeden robot do ďalšieho kola.

3. Analýza problému

Analýze riešeného problému a samotnej príprave na súťaž bolo venované veľa úsilia. Pred samotnou analýzou boli na základe pravidiel definovali nasledovné základné požiadavky, ktoré náš robot musel spĺňať pre súťažnú kategóriu Puck Collect:

- robot musel byť autonómny (schopný samostatného pohybu po hracej ploche),
- robot musel vedieť zbierať pukky,
- robot musel vedieť rozpoznávať farby zbieraných pukov,
- robot musel vedieť dopraviť čo najviac pukov vlastnej farby do domovskej zóny pred uplynutím časového limitu (3 min.).

Za účelom lepšieho pochopenia technických a programovacích techník robotov pre našu súťažnú kategóriu, sme sa zúčastnili robotickej súťaže RobotChallenge v roku 2009 ako diváci. Po analyzovaní súťažných robotov a svetelných podmienok, ktoré boli na súťaži boli vyvodené nasledovné závery:

- na súťaži panovali premenlivé svetelné podmienky, ktoré výrazne ovplyvňovali rozpoznávanie farieb zbieraných pukov.
- Robot, ktorý mal zakrytý senzor pre rozpoznávanie farieb pukov s osvetlením mal vyššiu úspešnosť rozpoznávania farby pukov. Ale aj tak niekedy došlo k nesprávnemu vyhodnoteniu farby puku.
- Roboty, ktoré sa vždy vracali do domovskej zóny po zozbieraní puku plytvali časom. (Jediná výhoda tejto stratégie bola v tom, že mali ako prvý puk v domovskej zóne. V prípade zhodného počtu pukov oboch súťažiacich vyhral ten, ktorý ako prvý priniesol puk do svojej domovskej zóny.)
- Zaujímavou bola myšlienka systematického prehľadávania celej hracej plochy pre jednoduchosť algoritmu. Nebolo potrebné riešiť rozpoznávanie pukov zo snímaného obrazu a bolo možné použiť menší výpočtový výkon.
- Roboty, ktoré systematicky prehľadávali plochu mali zásobník pre zbierané pukky.
- Spornou bola myšlienka ponechania si súperových pukov v zásobníku robota (pravidlá takúto možnosť nezakazujú.). Bolo to neférové voči súperovi. Výhodou tejto taktiky bolo to, že súťažiaci si zvýšil šancu na výhru.
- Triedenie pukov pomocou otočného triediaceho kríža sa nám javilo ako najlepšie hardvérové riešenie triedenia pukov.

4. Realizované riešenia

Na základe analýzy bolo rozhodnuté zrealizovať 2 súťažné stratégie:

- rozpoznávanie pukov,
- systematické prehľadávanie plochy.

Naše aplikácie boli naprogramované vo voľne dostupnej vývojovej platforme Eclipse [3] v programovacom jazyku Java s nainštalovaným Java Runtime Environment (JRE verzia 1.6) v 32 bitovom operačnom systéme Windows XP. Obe naše súťažné stratégie používali okrem iného aj všestranne použiteľnú zostavu Lego Mindstorms. Pri oboch súťažných stratégiách bola použitá NXT inteligentná kocka (obr. 5). Firmware NXT kocky bol nahradený operačným systémom leJOS [7], ktorý umožňuje interpretovať Java bytecode. NXT kocka je vybavená dvomi procesormi:

- hlavným 32 bitovým ARM7 procesorom (256 kB FLASH pamäť, 64 kB RAM pamäť),
- 8 bitovým koprocessorom (4 kB FLASH pamäť, 512 B RAM pamäť).

NXT kocka tiež disponuje bezdrôtovým komunikačným rozhraním Bluetooth (CSR BlueCore 4 v. 2.0). Ďalej pri oboch súťažných stratégiách boli použité nasledovné typy senzorov:

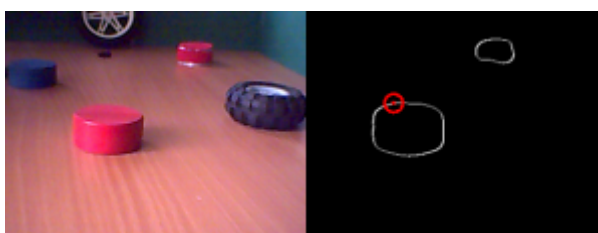
- HiTechnic color senzor - vyhodnocuje farbu snímaného objektu,
- 2 ultrasonické senzory - meria vzdialenosť od prekážky,
- HiTechnic kompas senzor - meria uhol natočenia robota,
- 3 NXT servo motory.

4.1 Rozpoznávanie pukov - 1. súťažná stratégia

Jadrom tejto stratégie bolo rozpoznávanie pukov z grabovaného obrazu z USB web kamery s nízkym rozlíšením. Po rozpoznaní nami zbieraného puku (buď modrého alebo červeného) bolo potrebné riadiť pohyb robota smerom k puku. Robotom boli zbierané všetky pukky (aj súperove). Ak bolo zistené, že daný puk je náš, bol uložený do zásobníka robota, inak sa puk prirodzeným pohybom robota dostal z neho von (obr. 6). Z dôvodov, ktoré sú uvedené neskôr bolo rozhodnuté pokračovať iba vo vývoji 2. súťažnej stratégie. Táto súťažná stratégia sa ďalej nerozvíjala.

4.1.1 Softvérová realizácia

Softvérová realizácia tejto stratégie vyžadovala 2 programy. Sekundárny program bežal v NXT kocke a hlavný program (riadiaci) bežal v notebooku (hlavná výpočtová jednotka). V notebooku bol spracovávaný snímaný obraz (rozpoznával pukky našej farby, vyhodnocoval ich polohu). Notebook posielal riadiace príkazy do NXT kocky. NXT kocka vykonávala riadiace príkazy, a okrem komunikácie s notebookom vykonávala autonómne riadenie otočného triediaceho križa (obr. 6), ktorý slúžil na triedenie pukov, ktorých farba bola už rozpoznaná.



Obr. 2: Snímaný obraz (vľavo) a obraz (vpravo) po aplikácii filtrov s určenou pozíciou najbližšieho červeného puku

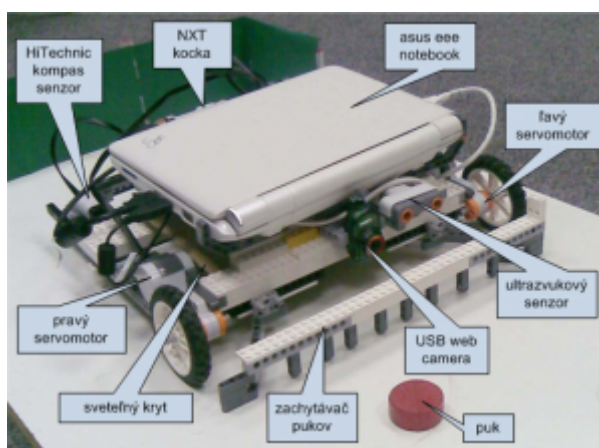
Princíp spracovania obrazu v hlavnej výpočtovej jednotke (notebooku) bol jednoduchý. Spracovanie obrazu spočívalo v postupnej aplikácii rôznych filtrov na zgrabovaný obraz z usb web kamery. Na zgrabovaný obraz sa najprv aplikoval filter Gauss blur [2], ktorý odstránil šum. Po aplikovaní tohto filtra bol aplikovaný zvolený farebný filter, ktorý odstránil všetky farby okrem danej farby (buď červenej alebo modrej).

Tento filter bol nami navrhnutý a vytvorený. Na základe H, S, B zložiek bolo možné filtru určovať rozsah hraničných hodnôt pre danú farbu. Po aplikovaní uvedených filtrov bol tiež aplikovaný Sobelov hranový filter [2], ktorý určil hrany jednotlivých

objektov z obrazu. Nakoniec bola aplikovaná Houghova transformácia [2], pomocou ktorej sa určila pozícia najbližšieho puku zvolenej farby (obr. 2).

4.1.2 Hardvérová realizácia - vývojová verzia robota

V tejto časti je ukázaná vývojová verzia robota (obr. 3) na ktorej boli testované obe súťažné stratégie. V prípade, ak bola testovaná 2. súťažná stratégia, z robota bol odobraný notebook a kamera kvôli odľahčeniu robota. Celý riadiaci program bol vykonávaný iba v NXT kocke. Postupne vývoj robota prešiel od použitia Gumstixu [4] a programovateľnej kamere CMUCAM3 [1] k zostave Lego Mindstorm NXT [6], asus eee notebooku a usb web kamere s nízkym rozlíšením. Robot sa prevažne skladal z NXT legových dielov kvôli ľahkej modifikovateľnosti.



Obr. 3: Vývojová verzia robota

Z analýzy vyplynulo, že svetelné podmienky výrazne ovplyvňovali úspešnosť rozpoznávania farby pukov. Preto bolo rozhodnuté, že oblasť robota so senzorom pre rozpoznávanie farieb bude zakrytá. Navyše, pridaním osvetlenia k senzoru pre rozpoznávanie farieb boli dosiahnuté svetelné podmienky. Tým bolo zamedzené vplyvu okolitých svetelných podmienok na správnosť rozpoznania farby puku.

Správne rozpoznávanie farieb pukov bolo jednou z kľúčových úloh, ktoré boli určené a výrazne zvyšovali naše šance v súťaži na úspech. Okrem toho bolo rozhodnuté použiť HiTechnic color senzor. Základný Light sensor, ktorý vedel určiť iba intenzitu svetla rôznej farby bol pre naše účely nevyhovujúci. Na súťaži sa nám počas súťažných kôl ani raz nestalo, že by náš robot zle rozpoznal farbu puku.

Pri testovaní tejto stratégie a robota bolo zistených viacero nedostatkov. Výkon motorov vzhľadom k celkovej hmotnosti robota bol nízky. Uhol záberu USB web kamery bol malý a kvalitu snímaného obrazu výrazne ovplyvňovali svetelné podmienky. Navyše, spracovanie obrazovej informácie bolo výpočtovo náročné. Preto bolo rozhodnuté, že použijeme iba 2. súťažnú stratégiu. Konštrukčným nedostatkom tejto realizácie robota bolo, že kompas senzor bol rušený vlastnými motormi robota.

4.2 Systematické prehľadávanie hracej plochy - 2. stratégia

Hlavnou a jedinou výpočtovou jednotkou bola NXT kocka. Ovládala otočný triediaci kríž, rozpoznávala farby pukov a ovládala pohyb robota tak, aby systematicky prešiel celú hraciu plochu a dokázal sa vrátiť späť do domovskej základne (domov).

Pri testovaní otočného triediaceho kríža bolo zistené, že sa občas tento kríž zasekol v dôsledku nesprávnej polohy puku pri otáčaní krížom. Táto situácia bola detekovaná na základe hodnoty z inkrementálneho snímača daného motora, ktorý ovládal otočný triediaci kríž. Ak sa motor mal hýbať a hodnota zo snímača sa nemenila, otočný triediaci kríž bol zaseknutý. Tento problém bol vyriešený tak, že po detekcii tejto situácie boli vysielané príkazy pre motor, ktorý ovládal otočný kríž, aby s ním pohyboval z boka na bok. Prirodzeným pohybom robota sa potom puk dostal do správnej pozície a problém bol vyriešený.

Po 3 minútach sa súťaž ukončila. Naším cieľom bolo, ako už bolo uvedené, dopraviť čo najviac pukov vlastnej farby domov. Bolo rozhodnuté, že robot bude 2 min a 10 s. systematicky prehľadávať plochu. Zvyšný čas bol využitý pre návrat robota domov. Robot nemal informáciu o tom, kde sa na súťažnej ploche nachádza. Vedel iba smer, ktorým sa pohyboval (buď smerom vpred, vzad, vľavo alebo vpravo). Ukážka smerov pohybu pre robota je na obr. 4. Pre jednotlivé smery pohybu mal robot definované uhli, na ktoré sa mal natáčať, keď sa robot otáčal.



Obr. 4: Smery pohubov robota pre červenú plochu.

Pri systematickom prehľadávaní plochy sa robot otáčal na základe údajov z kompasu senzora. Na súťažnej verzii robota (obr. 5) je kompas senzor umiestnený na stožiaroch. Tým pádom ho nerušili naše vlastné motory ako v prípade testovacej verzie robota (obr. 3). Počas súťaže bolo zistené, že motory niektorých robotov aj tak rušili náš kompas senzor. Táto situácia bola riešená tak, že pre NXT kocku boli korigované uhli pre natáčanie robota pre daný smer pohybu.



Obr. 5: Robot J2MP - pohľad z hora

Pri prehľadávaní plochy sa robot vyhýbal súperovmu robotovi. Po uplynutí uvedeného času pre prehľadávanie plochy, robot automaticky zahájil návrat domov. Pri tomto návrate sa robot už nevyhýbal súperovmu robotovi. Toto ošetrenie bolo riešené z dôvodu, kedy súperov robot môže byť v našej domovskej zóne. Mohlo by tak byť zabránené nášmu robotovi dostať sa do svojej domovskej zóny.

Robot sa snažil vždy pri návrate domov, prísť do rohu. Napríklad, ak náš robot štartoval z červenej zóny, tak pri návrate domov robot musel ísť najprv smerom vzad (obr. 4). Po detekcii prekážky (steny), sa robot otočil na smer doľava a šiel týmto smerom. Po detekcii prekážky sa ešte otočil na smer vzad. Ak opäť detekoval prekážku, tak bol v rohu (doma). Čiže robot sa dokázal úspešne navrátiť domov pomocou detekcie stien v priestore.

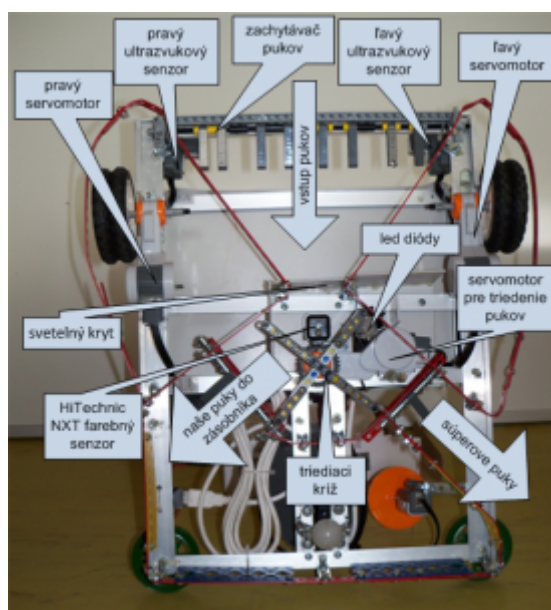
Pri zahájení návratu domov po uplynutí uvedeného času pre prehľadávanie plochy robot vždy chvíľu cúval. Tento úkon bol zavedený z dôvodu, ak by sa robot náhodou zasekol so súperovým robotom počas systematického prehľadávania plochy a nemohol by sa hýbať smerom vpred. Pri súťaži sa toto riešenie ukázalo byť ako kľúčové v 1 súťažnom kole, kedy sa náš robot zasekol so súperovým robotom. Náš robot sa nemohol hýbať vpred. Vďaka tomuto ošetreniu sa však náš robot dokázal vrátiť do svojej domovskej základne. Zároveň bola zistená chybná implementácia detekovania súperovho robota. Počas súťaže bol tento problém odstránený.

Ak oba senzory detekovali prekážku, tou prekážkou bola stena. Samozrejme, mohlo ísť tiež o súperovho robota. Bolo však rozhodnuté, že v tomto prípade budeme brať súperovho robota za stenu. Ak iba jeden z týchto senzorov detekoval prekážku, tak tou prekážkou bol súperov robot.

4.2.1 Hardvérová realizácia

Konštrukciu robota tvorili prevažne hliníkové profily a diely stavebnice Merkúr. Pomocné kolesá pomáhajú robotovi pri jeho otáčaní pri stene. Aj pri súťažnej verzii robota ošetrujeme vplyv svetelných podmienok na rozpoznávanie pukov pomocou svetelného kritu.

Keď sa puk dostane pod svetelný senzor, začne sa rozpoznávaním jeho farby. Puk pod senzorom vieme detekovať na základe toho, že farba podkladu je biela. Teda, keď sa puk dostane pod svetelný senzor, dôjde k zmene detekovanej farby z bielej farby na inú farbu. Po rozpoznaní puku by sa otočný križ otočil v protismere hodinových ručičiek, ak by puk bol takej farby, akej farby je domovská zóna robota.. Tým by sa puk dostal do zásobníka v zadnej časti robota. Inak by sa otočný križ otočil v smere hodinových ručičiek. Prírodným pohybom robota vpred by sa puk dostal z robota von. Keď puk opustil priestor otočného križa, tak mu vždy bolo zabránené, aby sa cúvaním alebo otáčaním robota dostal späť do tohto priestoru.



Obr. 6: Pohľad na robota zospodu

Pri otáčaní alebo pri cúvaní robota zohrával zachytávač pukov (časť robota na obr. 6) dôležitú úlohu. Nedovolil pukom, ktoré boli pozbierané a ktorých farba ešte nebola rozpoznaná opustiť priestor pod robotom pri cúvaní alebo otáčaní robota. Zároveň práve zbieraným pukom umožňoval vojsť pod prednú časť robota.

5. Záver

Ako J2MP tím sme sa zúčastnili súťaže RobotChallenge 20. 3. 2010 v kategórii Puck Collect v rámci tímového projektu. Pre našu súťažnú kategóriu boli zrealizované 2 súťažné stratégie:

- rozpoznávanie pukov,
- systematické prehľadávanie hracej plochy.

Kvôli viacerým nedostatkom 1. súťažnej stratégie bola dokončená iba 2. súťažná stratégia. Jej použitie na súťaži nám prinieslo 1. miesto v kategórii Puck Collect.



Obr. 7: Medaila za prvé miesto v kategórii Puck Collect

Poďakovanie

Slovenskú účasť na podujatí podporila Robotika.SK.

Literatúra

1. CMUcam3 – programovateľná kamera. Documentation. [online]. [cit. 3. 3. 2009]. Dostupné na internete: <<http://www.cmucam.org/wiki/Documentation>>
2. Computer Vision Demonstration Website. Electronics and Computer Science University of Southampton. [online]. [cit. 2. 12. 2009]. Dostupné na internete: <http://users.ecs.soton.ac.uk/msn/book/new_demo/>.
3. Eclipse. [online]. [cit. 14. 4. 2009]. Dostupné na internete: <<http://www.eclipse.org/downloads/>>
4. Gumstix Overo Earth. [online]. [cit. 14. 4. 2009]. Dostupné na internete: <http://gumstix.com/store/catalog/product_info.php?products_id=211>
5. Hitechnic. NXT sensors. [online]. [cit. 19. 11. 2009]. Dostupné na internete: <<http://www.hitechnic.com/products>>
6. Lego mindstorms. [online]. [cit. 2. 11. 2009]. Dostupné na internete: <<http://mindstorms.lego.com/eng/Overview/default.aspx>>
7. LEJOS. Java for LEGO Mindstorms. [online]. [cit. 2. 11. 2009]. Dostupné na internete: <<http://lejos.sourceforge.net/>>
8. Mindsensors.com. Sensors building kit for NXT with PCF8591 IC. [online]. [cit. 3. 3. 2009]. Dostupné na internete: <
http://www.mindsensors.com/index.php?module=pagemaster&PAGE_user_op=view_page&PAGE_id=92>
9. Oracle. Java Media Framework. [online]. [cit. 14. 4. 2009]. Dostupné na internete: <<http://www.oracle.com/technetwork/java/javase/download-142937.html>>
10. Parallax. The Boe-Bot Robot. [online]. [cit. 7. 4. 2009]. Dostupné na internete: <
<http://www.parallax.com/ProductInfo/Robotics/BoeBotRobotInformation/tabid/411/Default.aspx>>
11. RobotChallenge. Puck Collect. [online]. [cit. 24. 2. 2009]. Dostupné na internete: <<http://www.robotchallenge.org/reglement/puck-collect/>>

J2MP TEAM: Jozef Škultéty, Ján Maláč, Michal Beňo, Peter Mihál, Slovenská Technická Univerzita, Fakulta Elektrotechniky a Informatiky, Ilkovičova 3, 812 19 Bratislava, Slovenská republika
