УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП

Факултет за Информатика

Димитрија Ангелков

Веб базирана далечинска контрола на мобилни роботи

МАГИСТЕРСКИ ТРУД

Штип, септември 2013

Комисија за оценка и одбрана

Ментор: Доц. Д-р Сашо Коцески
Факултет за информатика, Универзитет Гоце Делчев Штип

Член: Проф. Д-р Цвета Мартиновска - Банде
Факултет за информатика, Универзитет Гоце Делчев Штип

Член: Проф. Д-р Андреа Кулаков
Факултетот за информатички науки и компјутерско инженерство (ФИНКИ)
Универзитетот „Св. Кирил и Методиј“ Скопје

Членови на комисија за оценка и одбрана:

Претседател: Проф. Д-р Цвета Мартиновска - Банде
Факултет за информатика, Универзитет Гоце Делчев Штип

Член: Доц. Д-р Сашо Коцески , Факултет за информатика
Универзитет Гоце Делчев Штип

Член: Проф. Д-р Андреа Кулаков
Факултетот за информатички науки и компјутерско инженерство (ФИНКИ)
Универзитетот „Св. Кирил и Методиј“ Скопје

Научно поле: Компјутерска техника и информатика

Научна област: Вештачка интелигенција и системи

Датум на одбрана: 30.09.2013

Датум на промоција: 30.09.2013

Веб базирана далечинска контрола на мобилни роботи

Веб базираната далечинска контрола е интересно и ветувачко поле за истражување во роботиката, виртуелната реалност и визуелизацијата. Апликациите кои се развиени во ова истражување ќе најдат примена во управувањето на роботите во фудбалскиот куп на роботи, вселената, подводната роботика, тренинг на операторите, медицината, едукација на далечина и насекаде каде имаме опасна средина во која човекот е изложен на ризик по неговото здравје. Во последните неколку години се направени многу системи преку веб базирана роботска контрола. Листата на активни системи контролирани преку веб пребарувачите секојдневно се зголемува. Негативните страни на овие системи се слабата реакција на системот од страна на управувачот кој е од клиентска страна. Целта на оваа магистерска работа е да креира систем кој ќе биде доволно брз и лесен за управување и кој ќе ги задоволи потребите на голем дел од операторите на системи кои се управувани на далечина. Денеска интернет технологијата дозволува на конвенционален начин да развиеме интегрирана мрежна околина за контрола на различни роботски системи. За да бидеме успешни во апликациите управувани во реално време, интернет базираните роботи имаат потреба од висок степен на слобода и автономност со локална интелигенција која ќе се справува со ограниченият проток и временските доцнења кои се последица од интернетот. Системот има стандарден мрежен протокол и интерактивен интерфејс човек-машина кој користи веб пребарувач за да може операторот на далечина на лесен и интерактивен начин да го контролира мобилниот робот. Со зголемување на роботите и роботските апликации се зголемува потребата да се најде начин преку кој тие ќе се контролираат на далечина. Контролата на роботите преку интернет е од особена важност поради тоа што постојат сигурни протоколи преку кои добиваме повратна информација и имаме интеракција човек и машина на далечина. Проблемот на далечинска контрола и управување на роботите, архитектурата на системи и интерфејси за контрола на еден или повеќе роботи во различни апликации е тема на постојано истражување во последната декада. Овој магистерски труд придонесува кон изнаоѓање на иновативни методи за управување на повеќе роботи истовремено во реално време преку интернет и дава одговор на три прашања.

Првото прашање е како да го направиме системот сигурен, второто прашање е како да го направиме системот брз и третото прашање е како да го направиме системот отворен и преносliв. За да го постигнеме тоа од клиентска страна е креиран графички интерфејс преку кој корисникот ќе го управува саканиот робот, а при тоа ќе има увид во содејството на сите останати роботи. Системот овозможува ексклузивна контрола врз роботите од страна на определени корисници. Секоја наша акција се запишува во базата и понатаму таа може да се анализира и врз база на нашите акции системот донесува анализа и соодветно поведение понатаму. Преку анализа на сликата од камерата која го следи движењето на роботите ги добиваме точните координати на роботите, потоа информациите се пренесуваат на серверска страна и се чуваат во база на податоци. Од таму постои друга апликација која ќе ги чита информациите и ќе ги прикажува визуелно самите роботи. На оваа апликација имаме опции за контрола на одбраниот робот со опција за ослободување на претходниот избран робот. Примената првично ќе биде во реализација на фудбалскиот куп на роботи, а понатаму и во секој производен процес којшто ќе има повеќе од еден робот.

Web-based remote control of mobile robots

WWW-based remote control is an interesting and promising research field in robotics, virtual reality and visualization. Applications that have been developed in this research will find application in the management of soccer robots, space, underwater robotics, training operators, medicine, distance education, and wherever we have a dangerous environment in which man is exposed to a risk to his health. In recent years, many WWW-based robotic control systems are created. List of active systems controlled through Web search engines grow every day. Disadvantages of these systems are weak response by the steering system that is client-side. The purpose of this master's work is to create a system that is quick and easy enough for management to meet the needs of many of the operators of the systems that are managed remotely. Today, Internet technology allows conventional way to develop an integrated network environment to control various robotic systems. To be successful in real-time applications run Internet-based robots require a high degree

of freedom and autonomy with local intelligence that will handle limited bendvit and time delays that are a result of the Internet. The system has a standard network protocol and interactive man machine interface that uses a web browser to remote operator can easy and interactive way to control mobile robot. With the increase of robots and robotic applications is an increasing need to find a way through which they will be controlled remotely. Control of robots via the Internet is of particular importance because there are reliable protocols through which we receive feedback and have interaction man and machine remotely. The problem of remote control and management of robots, systems architecture and interfaces to control one or more robots in various applications is a topic of constant research in the last decade. This master thesis contributes to find innovative methods for managing multiple robots simultaneously in real time via the Internet and provides the answer to the three questions. The first question is how to make the system reliable, the second question is how to make the system fast and the third question is how to make the system open and portable. To reach it from the client-side is created graphical user interface through which you manage the desired robot and when it will have an insight into the interaction of all the other robots. The system allows exclusive control over robots by certain users. Our every action is recorded in the database and it can be further analyzed and base our actions adopted and appropriate behavior analysis system further. Through the analysis of the image from the camera which follows the movement of the robots we get the exact coordinates of the robots then the information is transmitted to the server side and stored in a database. Since there is no other application that will read the information and will display visually robots themselves. For this application we have options to control the robot with the option chosen for the release of the previous selected robot. The application will initially be in the implementation of the Football Cup of robots, and later in every production process which will have more than one robot.

Содржина

1. ВОВЕД

Телероботските системи традиционално се дизајнирани и единствено управувани локално од човекот. Потребата од манипулација со механички уред на далечина датира од далечните педесетти години кога за прв пат се користи управување на далечина во нуклеарните централи. Потребата од управувањето со нуклеарното гориво е од особена важност бидејќи човекот се изложува на доза зрачење кое е опасно по неговиот живот. За да се елиминира присуството на човекот било потребно да се направи систем кој ќе се управува на далечина, и така настануваат почетоците на далечинското управување. Пионери во оваа област се Рајмонд Гоертз од националната лабораторија Аргон недалеку од Чикаго и Жан Вертут со неговите инженери од инженерската нуклеарна лабораторија во близина на Париз. Тие конструираат роботска рака со 6 степени на слобода која во себе има електромеханички и електро хидрауличен систем со актуатори и повратна врска до командната табла. На секоја зададена команда има повратна врска дали е успешно извршена. Целта им била за најкратко време да го префрлат радиоактивниот отпад од нуклеарната централа до канистерите за складирање на отпадот. Наредна визија им била да направат систем кој ќе се управува на далечина. Ваквиот систем кој го изградиле бил управуван до непосредна далечина. Потребата од елиминирањето на секоја последица од зрачењето била од особена важност и потребно било да се направи управување на далечина. Во оваа намера успеале да направат систем кој ќе функционира на далечина и ќе биде со мало временско доцнење од моментот на задавање на командите до моментот на извршување на командата. Во понатамошните години во времето од 1950 до 1960 година од минатиот век инженерите посакувале да направат телеробот кој ќе се управува на големи далечини, а целта им била да се направи робот кој ќе се управува на Месечината. Телероботот ќе дава видеосигнал од повеќе камери и врз база на таа информација управувачот од Земјата ќе дава соодветна команда. Во овој инженерски потфат пионери во 1960 година се Фарел од Институтот за истражување МИТ и Томсон од Универзитетот Стенфорд. Тие покажаа во се што сè треба да биде инволвиран човекот за да може да се управува робот на далечина. Подоцна доаѓаат на идеја системот да

има во себе автономна контрола во голем дел од операциите, а телеоператорот да има само увид во извршените и зададените команди. Ова е последица од временското доцнење на зададените команди и затоа најдоброто решение било да се креира систем кој има повратна врска на самото место. Тоа значи главниот програм за управување е на самиот телеробот, а не како пред тоа да е инсталиран во командниот центар.

Во последната декада интернетот прерасна во машинерија од комуникации која ги вклучи сите институции и индивидуи. Многу од корисниците на интернет сметаа дека тоа е систем кој се користи за праќање, примање на порака и отворање на веб-страници. Можностите на интернет се неограничени бидејќи постои огромен број на протоколи преку кои може да се оствари комуникација помеѓу машините и корисниците. Како пример: Се овозможува од работното место да пристапите до вашиот дом и да го контролирате преку интернет со соодветна интернет врска која во себе вклучува енкрипција и протоколи кои се сигурни и даваат неограничени можности во безбедноста на системот. Како и секоја нова технологија, а посебно управувањето на роботите на далечина е измислица во научната фантастика, и како што историски е запишано најпрво е користена во 1940 година за справување со радиоактивниот материјал. Денеска роботите се користат во најдлабоките океани и во вселената, да демонтираат бомби или да чистат опасни материи како што беше случај со хаваријата во нуклеарната централа во Фукушима, каде роботите беа од особена важност за справувањето со кризата. Почнувајќи од 1993 година новиот протокол HTTP доведе до развој на универзални веб-пребарувачи преку кои се отвораа сите страни кои беа хостирани на соодветните сервери, без разлика на кој оперативен систем беа извршувани. Овие пребарувачи правеа хипер линк до секоја од сликите и текстот во страната. Ова доведе до тоа да до 2001 година има инсталирано онлајн илјадници веб-камери. Доколку постоела можност една слика да биде линкувана, тогаш камерата не претставува ништо друго туку повеќе слики во една секунда поставени на серверска страна. Онлајн роботите најпрво се имплементирани за движење на камерите во посакуваната насока. На тој начин тоа резултира со задавање на команда во кој агол да биде поставена онлајн камерата која е управувана од клиентска страна. Од првиот онлајн телеоперативен робот кој беше активиран во август 1994 година до сега беа

поставени четириесетина онлајн телероботи од истражувачките тимови низ целиот свет. Онлајн роботите се присилени да користат механички уреди и интернет врска со слаб проток опкружени со непредвидлива работна околина во интеракција со човек кој ги задава командите и врши надзор на задачите на телероботот. Директниот телеоперативен интерфејс е заменет со интерфејс управување на далечина преку компјутер. Веб базираната контрола и едукацијата на роботи на далечина претставува растечко поле во инженерската едукација со суштинска вредност од едукативен материјал и многу алатки за едукација и платформи. Многу од универзитетите се адаптирани сопствените едукативни програми да ги изучуваат на далечина, а директната комуникација е заменета со новите технологии за пренос на видео, звук, датапринт и мултимедија. Професорот има интерактивна настава со голем број на студенти и со сигурност се остварува врска помеѓу предавачот и слушателот скоро идентична како на самото место. Денеска далечинското учење е имплементирано преку интернет и World Wide Web (WWW). Најиновативно во интернетот е тоа како за релативно кратко време прерасна во најмоќна компјутерска мрежа во светот. Во областа на контролата и роботскиот едукативен инженеринг имаме досега остварени важни резултати во развојот на интерактивни веб-базирани платформи кои продуцираат материјали за е. курсеви со онлајн теориски и лабораториски вежби кои се изработуваат преку веб-базирани физички лаборатории. Овие веб- базирани платформи може да се поделат во следните категории:

• Материјали за е. курсеви и е. училници;
• Виртуелни лаборатории;
• Далечински (физички) лаборатории;
• Комбинација од претходните три.

Од почетокот на цивилизацијата човекот имал замисла за креирање на суштество кое ќе личи на самиот човек. Општествата во раниот дел на првиот милениум ангажирале робови и ги користеле за да извршуваат задачи кои биле физички тешки и валкани. Робовите служеле за да ги хранат робовладетелите, за да го одржуваат нивното општество и да се концентрираат на тоа што го согледале како поважна задача како бизнисот и политиката. Човекот открил

механизми како на воденицата кои подоцна прераснале во роботи. Технолошкиот развиток бил бавен, имало покомплексни машини, генерално ограничени на многу мал број, кои извршувале пограндиозни функции. Електрониката сега станува водечка сила во развитокот на механиката со откривањето на првиот автономен робот создаден од Вилијам Грej Валтер во Бристол, Англија, 1948 година. Првиот дигитален и програмиран робот е откриен од Џорџ Девол во 1954 година и се вика Унимате. Од тогаш можеме да видиме посовремени роботи, како роботите АСИМО кои можат да одат и да се движат како луѓе. Роботите ги заменија робовите во извршувањето на тешките задачи, кои човекот преферира да не ги работи или се опасни по неговиот живот, како оние во вселената и на дното од морето каде човекот не може да ги преживее екстремните услови. Роботите се појавуваат во две форми: оние коишто можат да прават и да поместуваат предмети, како индустриските или мобилните роботи, и оние кои можат да се користат наместо луѓе, како андроидите на АСИМО и ТОПИО, и оние кои играат поспецифични улоги како Нано роботите и Рој роботите.

Од многу одамна луѓето се обидувале да направат машини кои ќе ги имитираат движењето и однесувањето на човекот и/или останатиот жив свет. Роботите може да се дефинираат како механички уреди кои може автоматски да изведуваат определени задачи. Во современиот дигитален свет, зборот робот многу често се користи за да означи и виртуелни софтверски агенти. Нема консензус за тоа кои машини се дефинираат како роботи, но има општа согласност помеѓу експертите дека роботите имаат тенденција да прават нешто, да се движат, да работат со механичките екстремитети, да ја осетат и манипулираат животната средина и да стекнат интелигентно однесување. Во зависност од задачите за кои се наменети тие имаат различна структура и комплексност, но најчесто се состојат од извршни уреди (актуатори), зглобови кои ги поврзуваат извршните уреди, сензори за перцепција на околината во која делуваат и процесорска единица која овозможува донесување на одлуки. Денешниот технолошки развиен свет не може да се замисли без нивното постоење. Роботите се применуваат секаде каде што се корисни. Како и луѓето, тие можат самостојно да извршуваат одредени задачи, но некои не се во состојба самостојно да ги извршат и имаат потреба од интеракција со човекот. За успешно да ги извршува зададените задачи роботот мора да биде

контролиран. Контролата на роботот се состои од три фази: восприемање, обработка и соодветно дејствување. Восприемањето на роботот се врши преку сензори. Сензорите му даваат информации на роботот за средината во која се наоѓа. Добиената информација мора да биде обработена, па во зависност од неа да се одреди дејството што ќе го преземе роботот. Обработката може да биде едноставна, а може да биде и комплексна. Кај случаите на едноставно обработување, обработувачот врз основа на сензорот дава директни наредби на извршниот уред, додека при сложените обработки има мешање на повеќе информации од различни сензори, па според тоа и покомплицирано решение. Големо внимание се посветува со помош на вештачката интелигенција роботот сам да изнаоѓа решенија дури и ако дадената ситуација не е предвидена во неговите наредби, односно да се направат роботи коишто ќе учат од средината, па според тоа самостојно ќе донесуваат решенија. По обработената информација останува уште извршните елементи да го извршат соодветното дејство односно наредба зададена од процесорот (мозокот) на роботот. Проучувањето на движењето на роботот се дели на две гранки, и тоа: кинематика и динамика. Директната кинематика ја проучува положбата на ефекторите во просторот, како и одредувањето на брзината и забрзувањето на ефекторите при познати позиции на зглобовите. Инверзната кинематика пак обратно при зададени вредности на ефекторите ја одредува позицијата на зглобовите. Директната динамика ја одредува брзината со која треба да се движат зглобовите и самите патеки кога се познати сите сили кои дејствуваат врз процесот. Инверзната динамика ја пресметува потребната сила за да се направат зададените движења. Роботите можат да се класифицираат според нивната специфична намена. Еден робот може да биде дизајниран да извршува една задача екстремно добро, или спектар на задачи многу подобро. Се разбира секој робот може да се репрограмира и да извршува различна работа од претходната, но сепак има роботи кои се ограничени од нивната физичка форма. На пример фабричката рака робот може да извршува неколку задачи, како: сечење, заварување, лепење и друго. Тоа се роботи кои можат да извршуваат повеќе задачи истовремено, независно од просторот и условите во него.

Во оваа магистерска работа за иницијално тестирање се употребени роботи од типот lego nxt2, истите се опремени со различни типови на сензори

и Bluetooth уред со соодветна софтверска апликација преку која ќе добиваат соодветни наредби преку комуникацискиот протокол. Камерата е високо над нив и снима, постојано го следи секое движење на роботите. Апликацијата која е на локален компјутер ќе ја анализира сликата и врз база на движењата на роботите ќе ги дава точните координати и ги препраќа на базата на серверската страна. Двете апликации комуницираат со базата и во зависност од брзината на серверот даваат соодветно дејство.

Со оваа магистерска работа се постигнува високо ниво на синхронизација помеѓу крајниот корисник и управуваните роботи. Целиот систем е и експериментално еволуиран, со цел да се добијат одговорите на определени фундаментални прашања, од типот: определување на времето потребно за обработка на сигналот од камерата и успешното праќање на информацијата во базата; анализа на времето од активирање на командата до нејзино успешно извршување на селектираниот робот кој се управува. Во поглед на препознавањето и следењето на роботите користени се два модула за препознавање на позициите на секој од роботите поединечно.

2. ПРЕГЛЕД НА СРОДНИ ИСТРАЖУВАЊА

За потребите на магистерскиот труд направени се три сродни истражувања на три типа роботи кои имаат можност да се управуваат на далечина. Особината да може да се управуваат на далечина е последица од инсталираните модули за комуникација кои ги имаат во себе. Овие модули ги читаат информациите од главниот управувачки компјутер преку кои имаат воспоставено сигурна конекција. Во еден од роботите комуникацијата е остварена преку Bluetooth, а во другиот е преку wi-fi конекција, додека во третиот тип на робот е комбинирана преку ултразвук, инфрацрвен и IRDA комуникациски модул. Ова е сè со цел да се види како системот ќе се однесува на различни комуникациски платформи и кои сè резултати ќе ги даде во однос на времето на примање и извршување на командите.

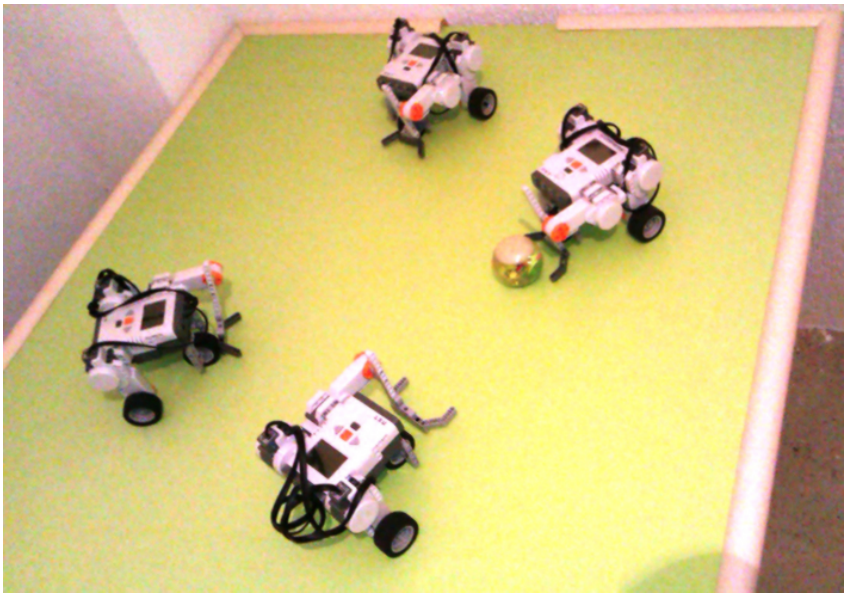
2.1 Lego роботи

Lego Mindstorms е линија на програмирани роботи/играчки, произведени од Lego Group. Доаѓа во кутија која содржи многу делови, сензори и кабли. Првата верзија од Lego Mindstorms излегла во малопродажба во 1998 година. Следната верзија била пуштена во малопродажба во 2006 година под името Lego Mindstorms NXT. Најновата верзија, пуштена во продажба на 5 август 2009 година е позната како Lego Mindstorms NXT 2.0. Хардверот и софтверот потекнува од Mindstorms Robotic, а главниот контролор е создаден од Media Lab. Првата визуелна програма, наречена LEGOsheets, била создадена на Универзитетот во Колорадо во 1994 година. Оригиналната Mindstorm кутија содржи два мотора, два сензора за допир и еден сензор за светлина. NXT верзијата има три преносни мотори и по еден сензор за светлина, звук и далечина. NXT 2.0 верзијата има по два сензора за светлина, звук и далечина. Mindstorm опремата исто така е користена и за едукација. Проектот е од соработката со LEGO и MIT Media Labaratory. Верзијата која е наменета за образование е наречена „Lego Mindstorms за училиштата“, и доаѓа со софтвер за програмирање. Но сепак, софтверот може да биде заменет и може да се користат и други програмски јазици, како на пример: Java или C. Единствената разлика помеѓу серијата за образование и серијата за потрошувачите е тоа што има уште еден сензор за светлина и повеќе опции за нагодување. Lego Mindstorms NXT е програмиран робот кој е направен од Lego во јули 2006 година, заменувајќи ја првата верзија „Lego Mindostorms“. Кутијата содржи 519 коцки, 3 мотори, 4 сензори (ултрасоничен, звучен, допир и светлина), 7 кабли за поврзување, USB кабел и NXT програмабилен контролор. Контролорот е познат и како „мозокот“ на Mindtorms машините. Тој му дозволува на роботот да прави разни комплицирани операции. Кутијата исто така содржи и NXT-G, графичка програма за програмирање која овозможува креирање и преземање на програми на NXT (главниот контролор). Lego Mindstorms NXT 2.0„Lego Mindstorms NXT 2.0“ е објавена на 5 август 2009 година. Кутијата содржи 619 коцки (вклучувајќи ги и сензорите и моторите), новиот сензор за боја, два сензора за допир и ултрасоничен сензор. За потребите на оваа магистерска

работа како задача е дадено да се направи робот кој ќе се управува преку интернет. За таа цел на располагање имаме робот Lego Nxt, веб-камера, компјутер кој ќе ја чита наредбата од интернет и сервис кој ја прикачува сликата од камерата на веб-страницата. Системот треба да работи беспрекорно во сите можни околности и да нема недостатоци. Со праќање и примање на потврдна команда се елиминираат сите можни нерегуларности кои може да настанат во системот во поглед на правилно и навремено извршување на командата. Склопувањето на роботот го правиме на следниот начин: земаме три мотори и со прави спојки ги прикачуваме на роботот. На секој од моторите ставаме тркала со гума и ги прицврстуваме со спојки на сервомоторите, потоа со кабли ги поврзуваме секој од роботите со лего коцката. Откога сите делови се поврзани го вклучуваме bluetooth на лего коцката. Потоа правиме стандардна конекција со мобилниот телефон. Конекцијата се обезбедува со креирање на конекциски пар т.е. преку внесување на лозинка со која се осигурува конекцијата и во секоја наредна конекција нема потреба од повторно осигурување на врската со помош на лозинка. Потоа, на третиот мотор ставаме рачка со која се придвижува топчето за да може да го шутира топчето кој голот. На слика под реден број 1 се прикажани роботите во работна верзија. Во оваа работна верзија е подготвен за движење и шутирање на топчето. Управувањето се врши преку интернет од посебно подготвена веб-страница преку која се задаваат командите. Со помош на инсталирана веб-страница сликата се прикачува на веб-страницата и на тој начин системот функционира. Овој функционален систем овозможува роботите да се подготвени за да играат фудбал. На слика 4 и слика 5 се прикажани деца како ги управуваат роботите во мало фудбалско игралиште. Преку два компјутера со bluetooth конекција се управуваат роботите во игралиштето. Во иднина ќе се направи големо игралиште на кое ќе играат повеќе играчи и ќе биде сето тоа онлајн т.е. на постојна веб-страница. Мрежната камера (или веб-камера од англ. Web-camera) е мала камера којашто се користи за пренос на видеоматеријални и правење слики коишто можат да се испраќаат преку интернет, а служат и за сметачки видеосостаноци. Мрежните камери се дигитални камери чии слики се испраќаат на некој мрежен опслужувач, постојано или на одредени интервали. Ова се постигнува или со користење на камери специјално направени за сметачи или пак со спојување на обичните

видеокамери со сметач. Може да се користат и аналогни камери кои се спојуваат со картичка за снимање на видео, а потоа сниманиот материјал се испраќа на интернет. Во магистерската работа се користи веб-камера со среден квалитет бидејќи има ранг во кој е дозволена промена на вредноста од пикселите со тоа програмот може да прима сигнал од различни камери со среден квалитет и дозволува со слаба опрема да се активираат успешно командите на секој од роботите. Традиционално, истражувањето од областа на мобилната роботика жестоко се насочени кон перцепцијата, моделирањето на околината и планирањето на патеката на движење. Како последица, возилата се дизајнирани да бидат компатибилни со ограничувањата при планирањето, а потребата од високи брзини е потисната. Притоа со подобрувањето на сензорските и резонирачки можности, постои опасност роботите да останат ограничени, не поради можноста за планирање, туку поради динамичките фактори. Роботите со тркала кои се способни за динамичко однесување, како на пример механизми за движење на лош терен со висока брзина, како и експлоатација на динамичката мобилност на возилата, претставуваат возбудлива и делумно истражувана област. Истражувачите најчесто ги гледаат мобилните роботи како квазистатички уреди. Бројни примери на роботи со четири, шест или повеќе тркала се развивани така да ја максимизираат мобилноста на лоши терени. За разлика од овие егзоскелетни роботи кои можеби имаат и поголем потенцијал за мобилност, се изградени и демонстрирани главно така да имаат ниско поставено тежиште и плоча како потпорна база, која заедно со интелигентен контролен алгоритам е дизајнирана да го задржува гравитациониот вектор на тежиштето низ базичниот полигон (т.е. нагибна моторика, ножна координација). Кај голем број концепти направен е обид за максимизирање на мобилноста со помош на големи тркала или нозе, со проширени обрачи на тркалата или стапалата на нозете, повеќетркален двигател, големо тело со голема површина и слободен простор со артикулирани конфигурации на телото. Ваквите роботи најчесто се ограничени во смисла на планирана мобилност, така што се дизајнирани за оперирање на мали брзини, кои типично достигнуваат до 1 км/ч или помалку. Динамичките фактори имаат мало влијание на ваквите системи, и поради тоа најчесто се игнорираат. За надминување на ваквите ограничувања во роботиката мора да се користат контролори со кои ќе се надминуваат постепено проблемите. За

потребите на овој магистерски труд изработено е соодветно игралиште со димензии 2 метра на 1 метар кое е прикажано на слика со реден број 1. Поставени се четири Lego NXT роботи кои се управуваат преку веб-страницата која има поддршка за php и mysql технологија. Основна задача на овие роботи е да си го преземаат топчето и да постигнуваат голови. Високо над нив ги снима IP камера која потоа сликата ја пренесува на веб-страницата. Видео стримингот може да се гледа директно онлајн и преку него имаме увид во секоја од акциите на роботите.



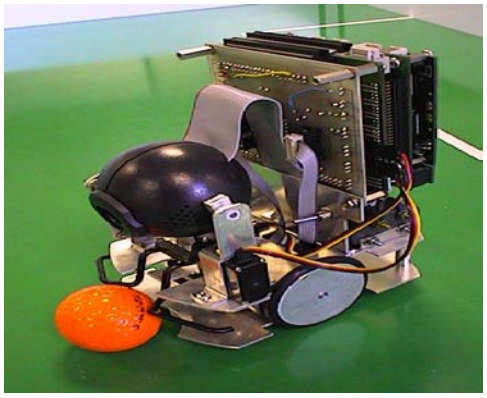
Слика 1. Четири роботи Lego NXT играат фудбал

Figure 1. Four Lego NXT robots play soccer

2.2 EyeBot робот

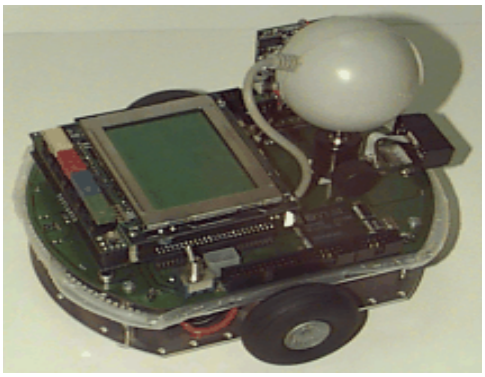
Еден од многуте контролори кои се користат моментално е EyeBot контролорот за мобилни роботи со тркала, роботи што одат или летачки роботи. Со овој контролор ни се овозможува да им ја зголемиме мобилноста и интелигентноста на мобилните роботи. EyeBot е контролор кој содржи моќен 32 битен микроконтролер со графички дисплеј и дигитална црно-бела или камера во боја. Прелиминарна цел беше да се креира платформа кај која ќе може камерата директно да се поврзи со платформата на роботот. Ова ни дозволува да пишуваме моќна програма за контрола на роботот без притоа да користиме

голем и моќен компјутерски систем и без да користиме поголем број на сензори со кои би си ја отежнале работата при пишувањето на програмот. Овој контролор се одликува со идеална база за програмирање и процесирање на слика во реално време. Во него има интегрирана дигитална камера (во боја или црно-бела) и голем графички дисплеј (LCD). Дава можност за проширување на множеството сензори и механизми за развивање на потполн мобилен робот. EyeBot е програмиран за IBM-PC или UNIX работна околина. Програмот се пренесува преку сериска линија (RS-232) во RAM или Flash Rom. Овозможено е програмирање во C или асемблер. EyeBot е контролор кој ја поддржува третата генерација роботи. На слика 2 и слика 3 е прикажан EyeBot.



Слика 2. Фудбалски агент Eyebot

Figure 2. Soccer agent Eyebot



Слика 3. Мобилен робот Eyebot

Figure 3. Mobile robot Eyebot

2.2.1 Апликациски програми

Gnu assembler (за PC и UNIX) или C компајлер (за PC или UNIX) можат лесно да се користат. Секако дека и други компајлери или асемблери за 68000 можат да се користат. Програмот во C може да се користи преку EyeBot libraries, преку кои се подготвуваат функциите за прикажување на излезите било на текст или графика, за влез на камерата (во боја или црно-бела), излез на звучник и контрола на сервомоторите. Функциите во Clib како „printf“ или „getchar“ можат да се користат како „LCDPutString“ или „KEY Get“ соодветно. Притоа мора да се внимава бидејќи тие ќе линкуваат до библиотеката со функции и ќе резултираат во подолга извршна програма.
За пример како да го програмираме во C е прикажано во наредните редови од програмот со наслов Demo.c

Пример 1, на еден C фајл

```
#include "eyebot.h"
#include <stdio.h>
int main ()
{ int k;
  printf("Zdaravo  !\n");
  k = KEYGet();
  printf("Pritisnato e  %d dugmeto !\n", k);
  return 0;
}
```

Потоа користиме команден ПРОМ за компајлирање на примерот.

gcc68 demo.cdl a.out

За да илустрираме преку еден пример како го компајлираме во асемблер, го користиме следниот програмски код:

```
    .include "labmac.i"
    .section .text
    .globl main
```

```
main:  LEA hello, A0          | Лаудирање адреса на стрингот
       CALLEXEC LCD_PutString | повик на асемблер рутина
       RTS
hello: .asciz "Zdravo korisniku !"
```

Потоа користиме команден ПРОМ за компајлирање на примерот. **gas68 hello.sdl a.out.** Можеме да користиме и комбиниран модел на C и асемблер. Користиме gcc68 –c за генерирање на индивидуални објект фајлови и потоа се користи Makefile за компајлирање на изворните фајлови.

2.2.2 RoBIOS

EyeBot користи RoBIOS v4.x оперативен систем. По поврзувањето на камерата и кабелот за напојување сè е подготвено за активирање на EyeBot. Напон на работа е 7.2V од извор на електрична енергија или со користење на батерии што се полнат. По активирањето ќе добиете коментар на графичкиот дисплеј.

```
- RoBiOS  Vx.x -
----------------
RobName 01 Cam:e
25 MHz  512K ROM
 896Kf 1024K RAM
 Battery-Status
@@@@@@@@@@@@@@@@@
<I> Hrd Usr Demo
```

EyeBot ја прикажува верзијата на оперативниот систем и големината на корисната меморија. Последната линија на текстот е четирибојни влезни копчиња (жолто, црвено, сино, зелено) оперативни како soft keys.

- **Info** за луѓето кои го користат EyeBot.
- **Hrd** поставување на параметрите и тест на камерата, сериската линија, сервата, моторите и останатите попратни уреди.
- **User program** повикување на програмот и активирање.

- **Demo** на функциите на камерата (црно-бела или во боја), звук и др.

EyeBot's operating system RoBIOS е предвиден за многубројни рутини со кои можеме директно да направиме повик до корисничките програми, а со тоа да не правиме компајлирање и повторно вметнување на програмот во EyeBot.
На почеток се почнува со хардверски рутини. Потоа влегуваме во хардверските поставувања и наидуваме на две менија.

Избираме

```
Hardware:


Set  HDT  IO
END
```

Под менито за поставување на параметрите

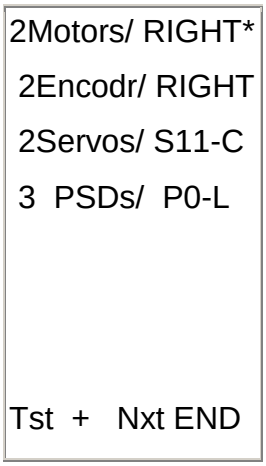
```
Setup:




Cam Ser Rmt END
```

Поставување параметри на камерата

```
CamSetup:

Bright :   140 *
Offset :    43
Contr. :   104
AutoBr.:   OFF
Version:   $00
 +   -   Nxt END
```

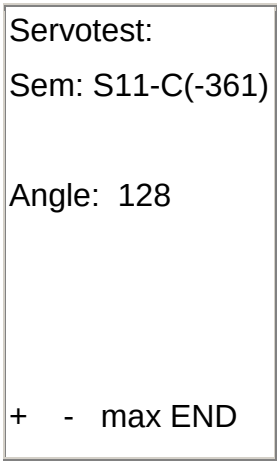
Тестирање на хардверските компоненти

Сите хардверски компоненти кои се дефинирани во HDT се дадени во листа овде и можат да се селектираат за индивидуални тестирања. Системот со два мотора, два енкодера, два сервомотори и три инфрацрвени сензори е покажано на сликата подолу



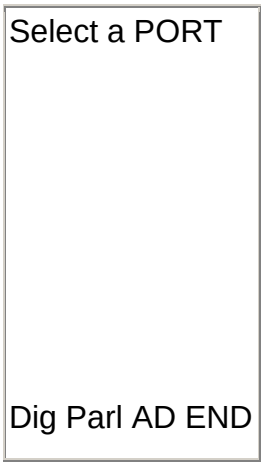
Индивидуални хардверски компоненти се селектираат со Tst, додека + прекинувачот помеѓу компонентите е без група, и Nxt прекинувачот помеѓу различните групи.

Независно од типот на селектираните хардверски компоненти, можеме да го активираме тест програмот. После селектирање на независен сервомотор можеме да го прочитаме позициониот агол и истиот да го контролираме.

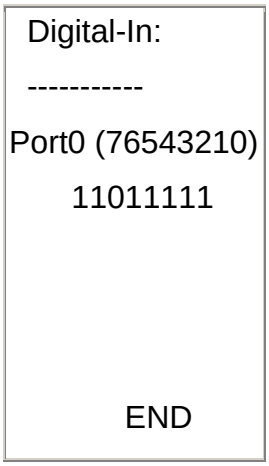


Тестирање на I/O портите

Менито за влезно излезните порти ни дозволува да го читаме и да го поставуваме поединечно бит по бит секој дигитален влез и секој излез било да е на паралелниот порт или на аналогниот влез.

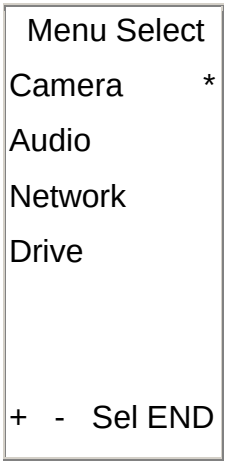


Пример со кој се чита статусот на сите влезни пинови .



Демо програмите се наоѓаат во ROM во „demos.hex“ и можат да се менуваат. Тие се активираат со притискање на „Demo“ копчето во главно мени и содржат демо програм за камерата, звукот, комуникациската мрежа и функции за движење.

Со притискање на „Sel“ се тестираат сите уреди.



2.2.3 Креирање на комплетен робот со EyeBot контролорот

Постојат повеќе начини за креирање на комплетен робот. Најпрво треба да се види каде ќе се користат сервомоторите (со кои ќе се служиме кај сите видови на движечки, летачки или роботи за сигнализација). Постојат многу различни цени на сервомотори. Треба точно да се утврди за каква намена ќе ни служи роботот. За роботите коишто играат фудбал се користат многу квалитетни сервомотори. За евтина реализација може да се користат модели на кола на кои ќе се имплементира EyeBot. Сензорите се следниот елемент во креирањето на комплетни мобилни роботи. Секогаш посакуваме да имаме дигитална камера кај сите типови на роботи, но во голем дел од нив се имплементираат сензори за детекција на растојание. Се користат инфрацрвени сензори за растојание, приближен инфрацрвен сензор, сонари, сензори на допир. Двоножните роботи што одат користат сензори кои во себе содржат компас за навигација. Оригиналниот Eyebot користи два мотора на прав напон со вградени кочници со енкодери, еден инфрацрвен PSD сензор, шест инфрацрвени сензори за приближно определување на растојание, акустичен систем, кочница и дигитална камера.

Слика 4. Eyebot мобилна платформа

Figure 4. Eyebot mobile platform

Поставеност на контролниот модул

TPU 0 right motor speed

TPU 1 left motor speed

TPU 2-3 left shaft encoder

TPU 4-5 right shaft encoder

TPU 2-13 servo channels (4 overlapping)
TPU 14 PSD
TPU 15 audio out

DO 0-1 left motor direction (A)
DO 2-3 right motor direction (B)
DO 4 PSD control line
DO 5-7 free

DI 0-5 PSD infrared sensor
DI 6-7 free (e.g. bumper)
AD 0 microphone
AD 1 battery level
AD 2-7 free

2.2.4 EyeBot андроид

EyeBot е збир од повеќе проекти за мобилни роботи. Сè се сведува на тоа да се користат што пофини сензори бидејќи располага со солиден развоен оперативен микропроцесорски систем. Механизмите и сензорската електроника се развивани индивидуално. Андроидите на сликата се високи по 50 сантиметри со 4 оптички сензори, два сервомотори во колковите и една дигитална камера во главата. Вкупно содржат 9 сервомотори со тоа претставуваат одличен пример за огромните можностите кои ги нуди овој систем. Овие роботи ни овозможуваат поблизок однос во релацијата човек машина.

Слика 5. Андроиди во кои е имплементиран Eyebot

Figure 5. Android which is implemented Eyebot

Слика 6. Андроиди во кои е имплементиран Eyebot

Figure 6. Android which is implemented Eyebot

Поставеност на деловите на Андроидот

(Поглед од позади)

```
              O       (camera as head)
              |
left arm   ----+---+---- right arm (Gen. II only)
           |   |
```

torso bend _/
left hip turn +---+ right hip turn
left hip bend | | right hip bend
left knee + + right knee
| |
left ankle + + right ankle (servos in Gen. I only, springs in GII)
* * (feet)

Johnny Walker

TPU 0 right hip turn
TPU 1 right hip bend
TPU 2 right knee
TPU 3 right ankle

TPU 4 torso

TPU 11 left ankle
TPU 12 left knee
TPU 13 left hip bend
TPU 14 left hip turn

TPU 15 audio out
other channels not used

Jack Daniels

TPU 0 right hip turn
TPU 1 right hip bend
TPU 2 right knee

TPU 4 torso
TPU 8 right arm
TPU 10 left arm

TPU 12 left knee
TPU 13 left hip bend
TPU 9 left hip turn

TPU 15 audio out

Останатите канали не се искористени.
За позиционирање на сервото од RoBIOS, еден бајт (0...255) е потребен.

value	**operation**
128	servo is in middle position robot is standing upright [arms pointing down]
0..127	move leg limbs behind the body move torso to the left turn hip joints counter-clockwise
129..255	move leg limbs in front of body move torso to the right turn hip joints clockwise

2.2.5 SoccerBot

Дизајнирањето на SoccerBot и SoccerBot Plus ни разоткри нови можности на управувањето во фудбалски Робокуп. Робокуп е интернационален куп на роботи кој е во потрага по одлични играчи кои се составени од тим со пет играчи. Купот на роботи може да се конфигурира како тим од единаесет играчи вклучувајќи и голман во тимот. Полето на активност на играчот се состои од фаќање, додавање и шутирање на топката. Контролорот се состои од два мотора на еднонасочна струја со вградена кочница и енкодер. Во себе вклучуваат стандардни механизми како и три инфрацрвени PSD сензори, LiIon батерии со полнач од 220 волти. Некои од нив како опција поседуваат и EyeNet безжичен комуникациски модул.

SoccerBot Plus во себе вклучува и проширен механизам со камера пан и механизам за шутирање, два сервомотора со метални кочници за камерата и операбилност на шутирачкиот механизам. Користат дигитална EyeCam камера и безжичен EyeNet комуникациски модул.

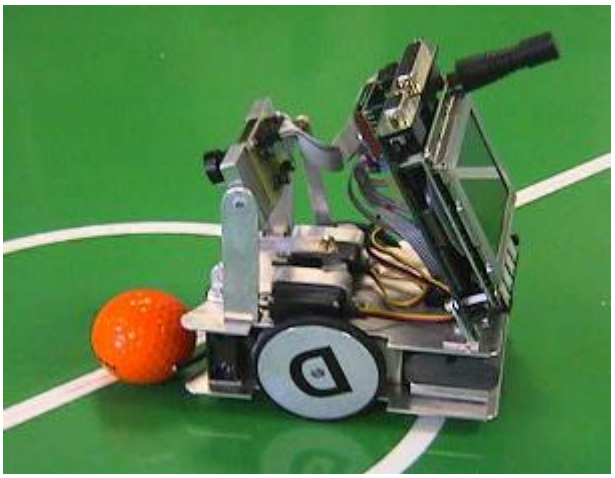
Поставеност на контролниот модул:

TPU 0 right motor speed
TPU 1 left motor speed
TPU 2-3 left shaft encoder
TPU 4-5 right shaft encoder
TPU 6-11 free
TPU 12 camera pan ("plus" only)
TPU 13 kicker ("plus" only)
TPU 14 PSD
TPU 15 audio out

DO 0-1 left motor direction (A)
DO 2-3 right motor direction (B)
DO 4 PSD control line
DO 5-7 free

DI 0-5 PSD infrared sensor
DI 6-7 free (e.g. bumper)

AD 0 microphone
AD 1 battery level
AD 2-7 free

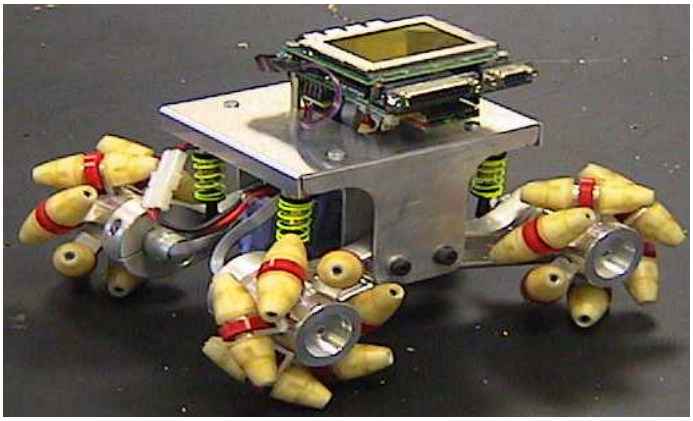


Слика 7. Фудбалски играч управуван преку wi-fi уред

Figure 7. Football player managed through wi-fi device

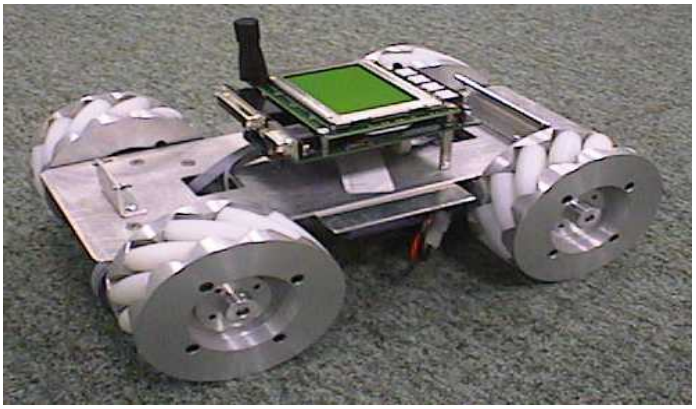
2.2.6 EyeBot возила

Следниот контролор се состои од четири независни погони со кои се овозможува точкасто движење во која било насока. Ова се должи на тоа што кружните делови во себе содржат ролери со кои се овозможува да се движи во секоја насока. Со тоа се дава супер подвижност на целиот систем. EyeBot контролорот одлично ја поддржува конструкцијата на ваквиот систем и во целост ја оправдува својата намена. Креирани се голем број на програми со кои се покажува неговата повеќедимензионалност и мултифункционалност. Omni е конвенционално дизајниран за движење. Тој во себе содржи ролери со кои стапува во контакт со површината. На сликата е прикажана неговата конструкција. Неговата подвижност се состои во тоа што на контактот од површина секогаш може да се придвижи ролерот кој е поставен под агол од 45 степени. Се состои од четири тркала со слободни ролери кои му овозможуваат голема подвижност на теренот.



Слика 8. Возила со вградени ролери на тркалата од надворешниот дел

Figure 8. Vehicles with built-in roller wheels of the outer part



Слика 9. Возила со вградени ролери на тркалата во внатрешниот дел

Figure 9. Vehicles with built-in roller with wheel implemented inside

Поставеност на контролниот модул:

TPU 0 front right motor speed

TPU 1 front left motor speed

TPU 2-3 front left shaft encoder

TPU 4-5 front right shaft encoder

TPU 6 back right motor speed

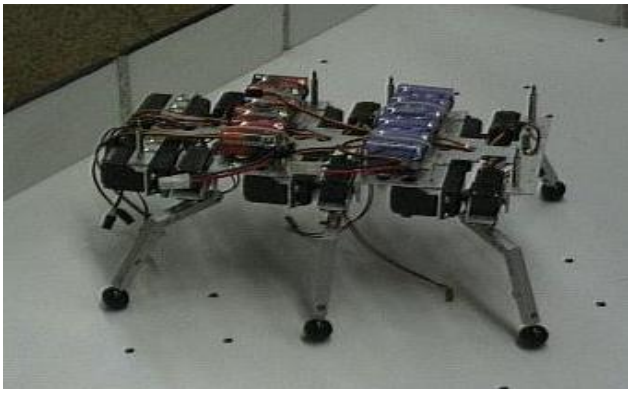
TPU 7 back left motor speed

TPU 8-9 back right shaft encoder

TPU 10-11 back left shaft encoder

DO 0,1 front left motor direction
DO 2,3 front right motor direction
DO -,5 back right motor direction
DO 6,- back left motor direction

EyeBot е збир од повеќе проекти за мобилни роботи кои во себе вклучуваат имплементација на голем дел од сензорите кои се наоѓаат на пазарот. Механизмите и електрониката на сензорите индивидуално се креирани од големиот дел на истражувачи кои се занимаваат со проблематиката на мобилните роботи. На слика со реден број 10, прикажан е еден робот кој се одликува со голема комплексност. Овој робот има во себе 12 сервомотори и два инфрацрвен сензори. Се одликува со негова лесна подвижност на тешки непристапни и непреодни терени.



Слика 10. Имплементација на Eyebot на мобилни роботи со шест нозе

Figure 10. Placing into six legs into Eyebot mobile robot

Спецификација на движачот
MK3 верзија (поглед од горе)

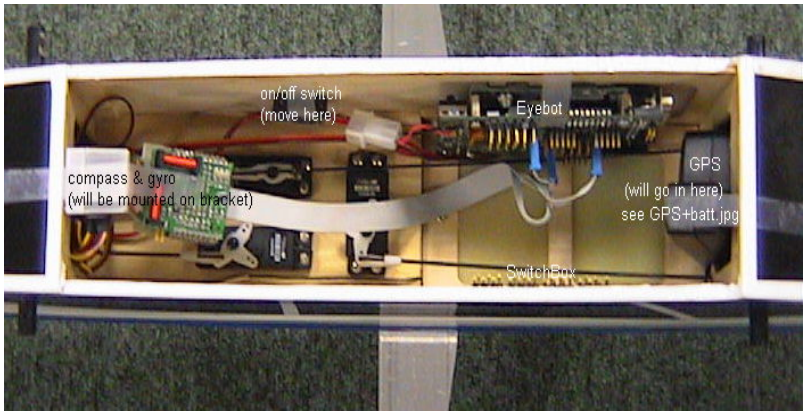
```
       / \ (глава)
нога 5 --+---+-- нога 6
       |   |
нога 3 --+   +-- нога 4
       |   |
```

нога 1 --+---+-- нога 2

Канал	Конектор	Функција
TPU 0-1	Motor A/B	не се користи
TPU 2-3	S1, S2	leg 1 лево/десно – напред/назад
TPU 4-5	S3, S4	leg 2
TPU 6-7	S5, S6	leg 3
TPU 8-9	S7, S8	leg 4
TPU 10-11	S9, S10	leg 5
TPU 12-13	S11, S12	leg 6
TPU 14	--	PSD infrared
TPU 15	--	audio out

2.2.7 FlyeBot

FlyeBot користи EyeBot controller и заедно со големиот број на сензори автономно врши навигација на авион. Во себе има вградено и систем за глобално позиционирање. Во себе има имплементирано програм за автоматска контрола на авионот. Тоа е овозможено бидејќи во себе има вградено жироскоп и дигитален компас. Со безжична комуникација се преминува од автоматски во рачен мод на работа.



Слика 11. Имплементација на Eyebot на летачки објекти

Figure 11. Placing Eyebot into flying objects.

TPU 2 Rudder

TPU 3 Elevator

TPU 4 Ailerons

TPU 5 Motor speed

DO 4 Сигнализатор на грешка

DI 0-7 компас

Serial1 кориснички (PC) интерфејс

Serial2 глобален позиционен систем GPS интерфејс

При усовршувањето на безжичниот комуникациски модул се вложува многу енергија и време, бидејќи тој е многу важен дел од комуникацијата со останатиот свет. Со соодветен плагин за контролорот е овозможена комуникација преку серискиот порт. Секој од EyeBot можат да комуницираат меѓу себе без потреба од централен компјутер. Вториот безжичен модул е развиен за комуникација со компјутер и преку него можат да се надгледуваат роботите. Работна фреквенција е 433 MHZ со рата на пренос од 9600 Baud, со протокол за детекција на грешка и автоматска конфигурација на мрежата. Инфрацрвениот сензор PSD (Sharp) директно се поврзува на EyeBot. Дигиталниот компас т.е. Vector Compass 2X се поврзува на серискиот порт.

Compass module Other D-InOut Eyebot

XFLI NC - (or GND for invert)

YFLI NC - (or GND for invert)

BCD NC -

RST PIN 8

MS GND -

SS NC -

CI NC -

ECO NC -

RAW NC -

GND - PIN 15,16

5V - PIN 13,14

CAL INVERTED PIN 7

PC INVERTED PIN 6

RES NC -

SD1 NC -

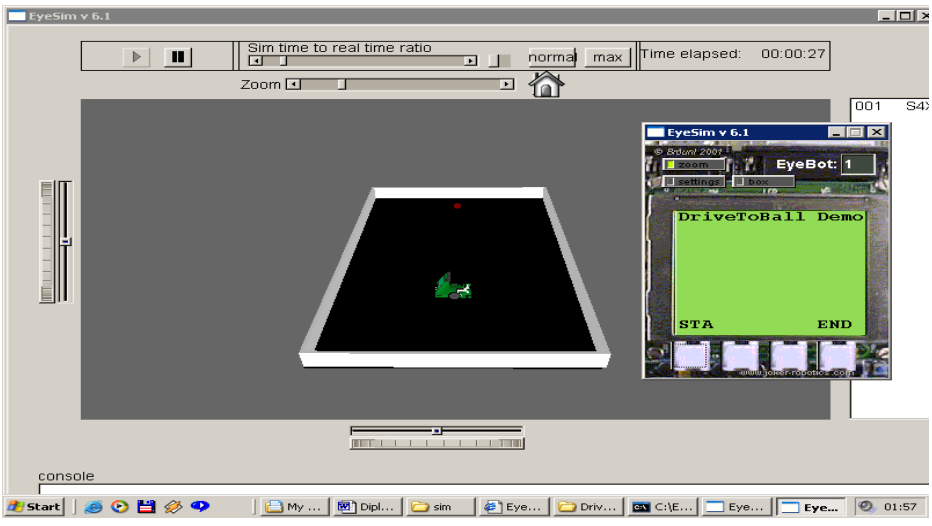
SD0 - PIN 9

SCLK - TPU-X (set in HDT)

1 X 100n BYPASS CAP BETWEEN 5V & GND

2.2.8 EyeSim

EyeSim програмот е симулатор за повеќе роботи кој ни дозволува да експериментираме без да ја менуваме програмата во мобилниот робот. Во себе содржи симулација на сликата од камерата на движечките актуатори и сите пропратни сензори кои се составни елементи на мобилните роботи, како што се амортизерите, инфрацрвените сензори, одометрите и останати сензори. Со EyeSim имаме 3D репрезентација на сцената во работната околина на роботот и сите околни роботи. Тоа најдобро се гледа на сликата под реден број 11. Со кликање на саканиот робот добиваме негов еквивалент на LCD дисплејот и неговите копчиња за комуникација на роботот со корисничкиот интерфејс.



Слика 12. Симулационен програм за наоѓање на топка во простор

Figure 12. Simulation program for finding the ball in working space

За симулација едноставно повторно ја компајлираме програмата користејќи ја библиотеката на RoBIOS која е корисничка библиотека на роботот. Потоа креираме друга верзија со идентичен наслов која ќе ни служи за симулацискиот програм. Со тоа работата ни се поедноставува.

2.2.9 EyeSim особини

- За да компајлираме најпрво правиме линк до симулациската библиотека.
- gccsim DviziSe.c
- Симулација на сите функции на RoBIOS.
- Идентичен модел на грешки за сензорите, моторите и квалитетот на сликата.
- Влез за копчињата и LCD излез.
- Синтетичка слика во боја за секој робот посебно (со респективен поглед за секого од нив).
- Повеќе роботска симулација.
- Дозволено е навигирање на роботот во симулационата околина.
- Симулациони сензори: придушувачи, PSD и IR-proxy.
- Конфигурацијата на сензорите и нивните вредности се прикажани на дисплеј.
- Симулациони актуатори на v-omega со пренос на движење.
- Simulated actuators: v-omega differential driving
- Конкурентност е во тоа повеќе роботи да можат да се симулираат во реално време во истата околина (во тој случај да не се користат глобални променливи).
    - Секој од роботите можи да има независен број на конкурентни тредови.
    -
- Параметарски фајлови
    - Fajl.sim

      Главен фајл за симулација со кој се линкува до околината и фајловите на роботот.

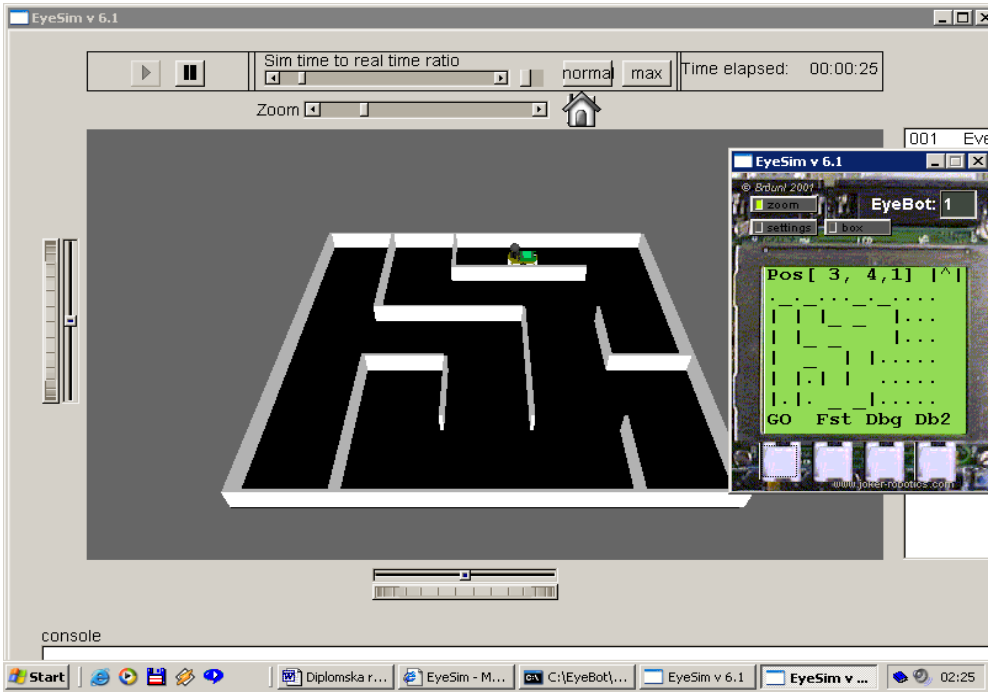
- MojotRobot.robi

  Опис на роботот со физичките димензии и локацијата на сензорите.
- MojotRobot.iv file

  OpenInventor 3D графички опис на роботот.
- Eyesim.ini file

  Фајл за иницијализација на симулацијата, како и параметрите за времето и поставеност на грешките и др.
- mojataOkolina.maz

  Поставеност на околината .



Слика 13. Симулационен програм за движење низ лавиринт

Figure 13. Simulation program to move through the maze

На сликата со реден број 13 е дадена симулација на мобилниот робот во лавиринт. Неговата задача е да даде приказ на лавиринтот на графичкиот дисплеј. Тој се движи во сите негови делови и ги регистрира препреките и откога ќе го помине целиот лавиринт се враќа на почетната позиција. Во почетната позиција ни го дава добиениот приказ на лавиринтот. Потоа му задаваме координата во лавиринтот за цел и тој ни го исцртува зададениот пат

на дисплејот. Со четирите копчиња го поставуваме режимот на работа и со тоа во целост ја доловуваме вистинската претстава за целокупниот систем кој иако е симулација доста верно ни ја претставува оригиналната работна околина. Целта пред на овој труд е да се користат евтини мобилни роботи кои ќе бидат достапни на располагање за секој истражувач. Тие нема да претставуваат пречка во никој поглед на неговиот истражувачки дух бидејќи со постојните симулациони програми не се остава никаков елемент на обесхрабрување. Сè се сведува на развивање на добар алгоритам во C или асемблер за потоа симулаторот успешно да ја тестира нашата програма. Иницијативите на Светскиот фудбалски куп има намера да ги поттикне истражувачите во роботиката и вештачката интелигенција и со тоа да обезбеди место каде што повеќе технологии ќе можат да се обединат. Затоа организацијата на робокуп го избра фудбалот и организира куп таканаречен: Robot World Cup Soccer Games. За да може еден тим од роботи да игра фудбал, мора да се вклучат повеќе различни технологии: дизајн од автономни роботи, соработка меѓу повеќето агенти, стратешки набавки, навремено реагирање, роботика и комбиниран сензор. Робокупот е натпревар меѓу типови од брзо движечки роботи во динамична околина. Тој нуди и софтверска платформа за испитување на софтверските аспекти на робокупот. Во историјата на вештачката интелигенција и роботиката 1997 година ќе се запамети како година на пресврт. Во мај 1997 година IBM Deep Blue го победи светскиот шампион (човек) во шах. Четириесетгодишниот предизвик со заедницата на вештачката интелигенција дојде до успешен заклучок. На 4 јули 1997 година Мисијата на NASA за пронаоѓање патеки,успешно се приземји на Марс и првиот автономен роботски систем Sojourner се прошета по површината на Марс. Заедно со овие достигнувања, Робокупот ги направи првите чекори кон развојот на роботите фудбалери. Робокупот беше основан од компјутерската лабораторија во Токио во 1997 година. Целта се состоеше во тоа да се развие фудбалски тим кој ќе го победи светскиот шампион во фудбал. Достигнувањата не изгледаа баш изводливи во тој момент, со постојната технологија тоа е возможно. Сепак најголем дел од неговите потцели се реализираа и успешно најдоа примена во голем дел од техниката.

Слика 14. Фубал со агенти од различни платформи

Figure 14. Fubal with agents from different platforms

Најголема примена на EyeSim и EyeBot доживеа во натпреварите со фудбал каде сите негови доблести се покажаа на најинтересен и најзабавен начин. Се посвети голем труд на машинскиот дел. Се воведоа посебни стандарди за голмани, напаѓачи и одбранбени играчи. Платформата која беше дотогаш развиена сосема ги задоволи потребите на развиените алгоритми и придонесе до негов голем развој. Сè беше решено со постојниот микроконтролор на моторола M68322 кој директно се вметнуваше во платформата и директното поврзување со камерата која е од суштинска важност при детекцијата на динамичката топка. Надворешни процесирања не беа потребни и тој се покажа како доволен за сите операции коишто програмерите можат да ги додадат во овој оригинален и за првовремено многу надежен платформски систем. Славниот CIIPS роботски систем се состоеше од пет играчи и еден голман. Секој од играчите имаше по седум инфрацрвени сензори со кои приближно го определуваше растојанието до соседниот робот. Четири од напаѓачите имаа различни улоги во зависност од местоположбата. Во секој робот е вграден сензор со амортизер со кој се детектира колизијата помеѓу роботите. Секој од роботите има 24 битна камера во боја со директно поврзаност до роботската платформа. Во платформата се вградува и програмски сензор кој му користи за навигација. Sидовите и головите имаат посебна боја, топката е црвена. Сето ова се прави за полесно да се снаоѓа мобилниот робот во фудбалскиот терен. Привремено се

развиваа едноставни алгоритми за да со тек на време Робокупот со своите роботски платформи на EyeBot прерасне во светски шампионат. За детекција на топката и движењето помеѓу играчите се развиваа безброј алгоритами кои успешно се тестираа на симулаторот, а потоа и на теренот каде се покажуваше задоволителната симулација на реалните проблеми. Ограничени од слабата процесирачка моќ се бараат едноставни и ефективни решенија. Брзата анализа на околината и земањето на попрецизни резултати од пресметката се основните таскови за секој играч. Роботите мораат секогаш коректно да ги следат промените на околината. Специфичната боја на топката која е портокалова (понекогаш и црвена) и различната боја на двата гола (сина, жолта) се влезни факти кои роботот ги зема во себе пред секој натпревар. По почетокот на играта се анализира секоја новодобиена слика од колор камерата. Ако се детектира топката тоа се глобални координати кои се калкулираат од позицијата на ново пристигната слика. Во зависност од позицијата на топката и позицијата на роботите на теренот играчот ја одбира соодветната операција за движење и ја извршува. Независниот систем му овозможува на секој од роботите да ги контролира своите сензори со тоа што во зависност од акцијата ги вклучува или исклучува на соодветен начин. Уште поважно од тоа е, не само да реагираат туку и да одговорат на специфичната ситуација на фудбалскиот терен без многу двоумење. Стратегијата на играта игра многу важна улога во симулационата лига каде единаесет виртуелни играчи играат едни против други. Постојат повеќе категории од кои поважни се учењето и тимската работа. Основен проблем е тоа како да се научи тим од софтверски агенти да се бори и да постигне цел, во атмосфера на соработка од една страна, а против другиот тим од роботи кој се бори за истата цел од спротивна страна. Има неколку начини на учење на роботите: индивидуален OFF-LINE учење на вештини, OFF-LINE соработка и учење преку тим ON-LINE вештини, учење преку соработка и ON-LINE неповолно учење.

Тимската работа е од особена важност за правилно избраната стратегија во текот на играта. Во централно управуван тим од роботи тимската работа мора да се планира и изврши во вистинското време. Особено е важно да се исклучат сите несигурности и непредвидени промени. Ова вклучува динамички промени во целите на тимот, членовите на тимот и нивната неочекувана неспособност да ги исполнат своите обврски или пак неочекувано откритие на нови

можности. Важно е главната стратегија на тимот да не биде заснована на претходно планирана соработка, која најчесто потфрлува. Тимската работа се состои од планирање за повеќе роботско играње (дефинирање на можните стратешки цели и нивно достигнување), поединечни планови и нивно соединување и извршување на глобалниот план на тимот. Програмите се проценуваат со акција на основната изведба на тимот, снагата наспроти промените, генералниот перформанс, временските реакции и учењето. Друг важен аспект се акциите кои ќе може роботот да ги изведува. Како резултат на ова се јавува интерес да се контролира топката во повеќе роботен систем. Од ден на ден сè повеќе се зголемуваат подвижностите и нивната опционалност при манипулација со топката. Со ова им се овозможува вклучување на 3D контролирање на движењето на топката и играње со роботи кои имаат контрола на чекорењето. Потребно е да се изберат различни однесувања кои зависат од актуелната состојба. Бидејќи целта е фудбалскиот тим да продуцира неедноставни однесувања кои се доволни од нив да произлезат интелигентни поведенија. Само со разбирање на состојбата во целата нејзина комплицираност (кое нешто е оневозможено во овој момент, но се постигнува до одреден степен) може да се одбере соодветната акција, во однос на позицијата на сите роботи на теренот. Поради многуте ситуации подобар пристап е можноста роботот да учи, отколку да се обидува да ги предвиди сите можности. Најголемата тенденција на мојата работа се заснова врз ефикасно и рутинско процесирање на сликата. Роботите мора да бидат едноставни, за да бидат брзи. Мора да бидат и доволно моќни за да детектираат објекти во слики со ниска резолуција. Во мојата програма процесирањето се засновува само на детектирање на боите. Сликите кои се користат имаат резолуција 80 x 60 пиксели и се менуваат околу 2,5 пати во секунда. Втората цел е да се пронајдат неколку нови рутини на возење. Тие ќе му дозволат на роботот да ги достигне своите цели без поголеми девијации, како и за процесирање на сликите заради временското ограничување на калкулација. Мора да се извршуваат што е можно најбрзо. За правилно детектирање на топката во игралиштето го користиме подолу дадениот програмски код во кој во коментарите е опишан и објаснет секој ред од програмот. Целиот програмски код се засновува на препознавање и следење на соодветна боја. Топката е со специфична боја и затоа програмот лесно ќе ја препознава.

2.2.9.1 Програм за детекција на топката во фудбалското игралиште

Во овој програм ќе илустрираме програмски код кој ќе врши препознавање на топката во игралиштето. Топката има специфична боја и програмот врши нејзино препознавање и следење на истата. Сè додека не ја постигне зададената површина на кругот од сликата не престанува со движење кон неа.

```
#include "eyebot.h"
#include <stdlib.h>
#define MIN(a,b) (a<b?a:b)
#define MAX(a,b) (a>b?a:b)
#define NO_HUE -1
#define BALLHUE 41
int RGBtoHue(BYTE r, BYTE g, BYTE b)
/* враќа вредност за RGB боите кои ги детектира */
{ int hue, delta, max, min;
  max   = MAX(r, MAX(g,b));
  min   = MIN(r, MIN(g,b));
  delta = max - min;
  hue =0; /* иницијализација на боја*/
  if (2*delta <= max) hue = NO_HUE; /* нема боја */
  else {
    if (r==max) hue = 42 + 42*(g-b)/delta; /* 1*42 */
    else if (g==max) hue = 126 +42*(b-r)/delta; /* 3*42 */
    else if (b==max) hue = 210 +42*(r-g)/delta; /* 5*42 */
  }
  return (BYTE) hue; /* сега вредностите се движат од  [0..252] */
}
void ColSearch(colimage img, int obj_hue, int thres,
               int *pos, int *val)
/* ја бара x позицијата на обоениот објект, враќа pos и вредност соодветна*/
{ int x,y, count, h, distance;
  *pos = -1; *val = 0;  /* init */
  for (x=0;x<imagecolumns;x++)
  { count = 0;
```

```
    for (y=0;y<imagerows;y++)
    { h = RGBtoHue(img[y][x][0],img[y][x][1],img[y][x][2]);
      if (h != NO_HUE)
      { distance = abs((int)h-obj_hue);  /* hue distance */
        if (distance > 126) distance = 253 - distance;
        if (distance < thres) count++;
      }
    }
    if (count > *val) { *val = count; *pos = x; }
  }
}
int main()
{ colimage c;
  VWHandle vw;
  int hue, pos, val, key;
  CAMInit(NORMAL);
  vw = VWInit(VW_DRIVE,1);
    LCDPrintf("Se dvizam do topkata\n");
  LCDMenu("STA","","","END");
  key = KEYGet();
  while(key != KEY4) /* додека не е притиснато копчето четири */
  {       LCDMenu(" ","","","  "); /* чисто мени  */
    do
    { CAMGetColFrame(&c,0);
      LCDPutColorGraphic(&c);
      ColSearch(c, BALLHUE, 10, &pos, &val);  /* бара слика */
      LCDSetPos(1,0);
      LCDPrintf("h%3d p%2d v%2d\n", hue, pos, val);
      if (val < 20)  /* во друг случај заврши  */
        if (pos == -1 || pos < 20)  VWDriveTurn     (vw,   0.10, 0.4);  /* се придвижува
лево     */
         else if (pos > 60)             VWDriveTurn     (vw, -0.10, 0.4);  /* се придвижува
десно
           else                         VWDriveStraight(vw,  0.05, 0.1);  /* се движи напред
```

```
                                    VWDriveWait(vw);  /* привршува со движењето */
  }          while (val < 20);
                                /* прашај за следна детекција */
     LCDMenu("STA","","","END");
    key = KEYGet();
  } return 0 ; }
```

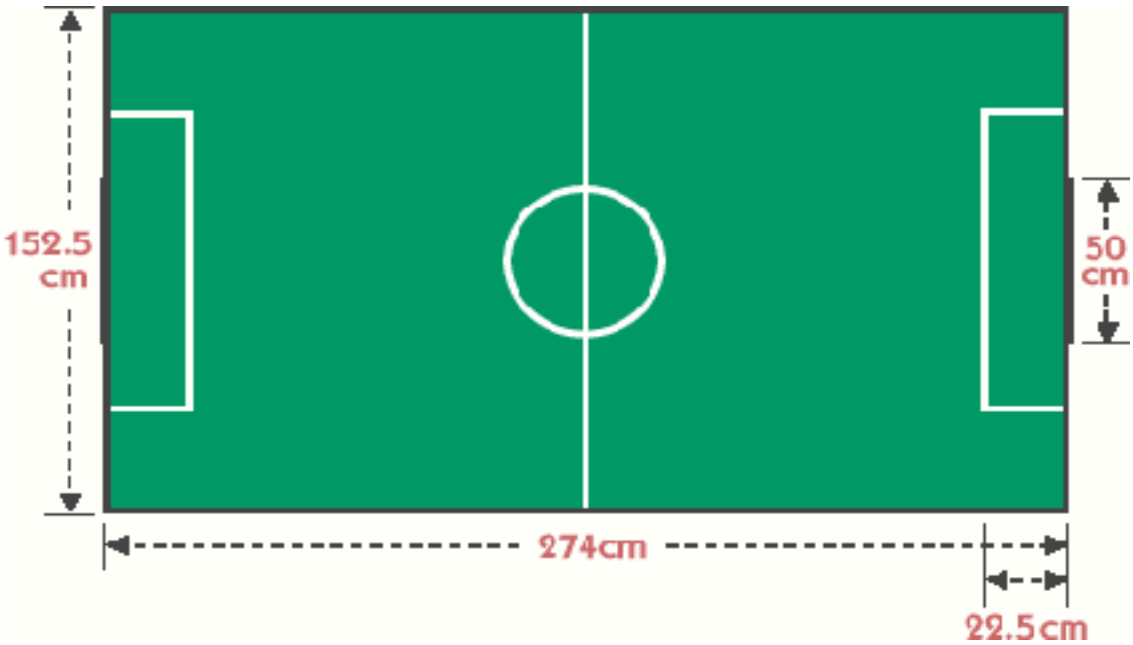
На сликата под реден број 15, е прикажано практична симулација на погоре наведениот код.

Слика 15. Детекција на објект во движење

Figure 15. Detection of objects in motion

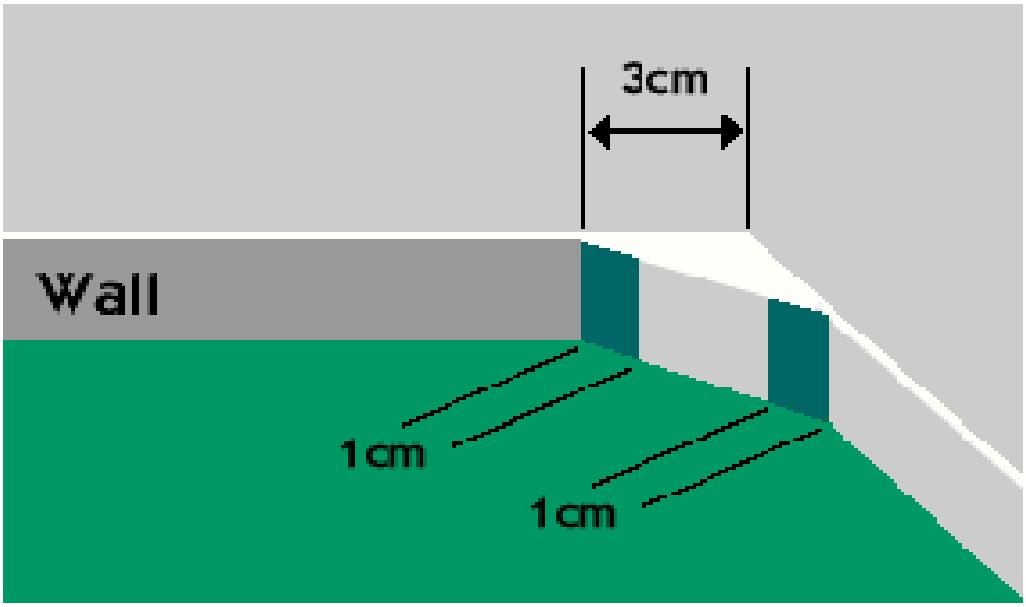
За наједноставен пример од кој секој ентузијаст и истражувач тргнува е најпрво решавање на алгоритамот за произволно слободно движење на роботот. Во подолу дадениот код како за пример се гледа неговата едноставност, така да можеме прво времено да креираме почетен тим кај кој секој од агентите произволно ќе се придвижува по теренот соодветно и за другиот тим. Со тоа сме изградиле фудбалски тим кој е подготвен за понатамошен развој. На сликата се дадени димензиите на фудбалското

игралиште кои ни се од особена важност за правилно дефинирање на параметрите во контролите на слободното дејствување на агентите.



Слика 16. Стандардни димензии на мало игралиште

Figure 16. Standard sizes on small playground



Слика 17. Стандардни димензии на мало игралиште

Figure 17. Standard sizes on small playground

2.2.9.2 Програмски код за слободно движење

Во овој програмски код ќе покажеме едно слободно движење на роботот чисто за демонстрација и тестирање на самиот робот.

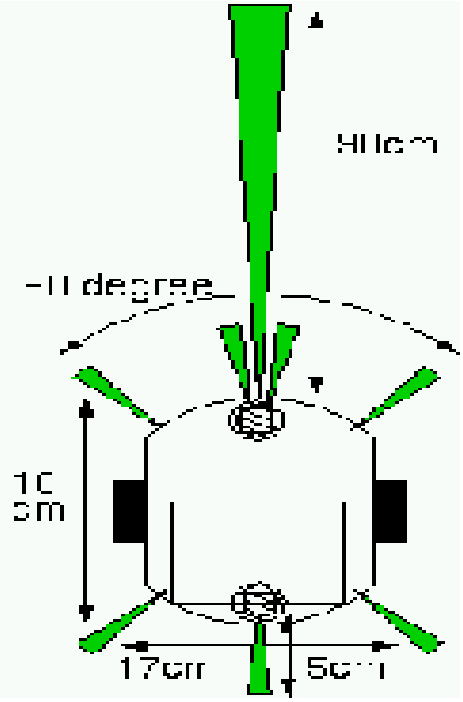
```
#include "eyebot.h"
#include <stdlib.h>
#define M_PI 3.14159265358979323
#define SAFETY 300
int main ()
{ PSDHandle front, left, right;
  VWHandle  vw;
  float     dir;
  colimage  img;
  LCDMenu("", "", "", "END");
  vw=VWInit(VW_DRIVE,1);
  VWStartControl(vw, 7.0,0.3,10.0,0.1);
  front = PSDInit(PSD_FRONT);
  left  = PSDInit(PSD_LEFT);   right = PSDInit(PSD_RIGHT);
  PSDStart(front | left | right , TRUE);
  CAMInit(NORMAL);
  while(KEYRead() != KEY4)
  {        CAMGetColFrame(&img,0);  /* само за демонстрација  */
          LCDPutColorGraphic(&img);
    if ( PSDGet(left) >SAFETY && PSDGet(front)>SAFETY
&&  PSDGet(right)>SAFETY && !VWStalled(vw) )
     {       VWDriveStraight(vw, 0.5, 0.3);  }
    else {    VWDriveStraight(vw,-0.04,0.3);
                VWDriveWait(vw);
  dir = M_PI * ((float)rand()/(float)RAND_MAX - 0.5); /* произволен агол -90 .. +90 */
                VWDriveTurn(vw, dir, 0.6);
                VWDriveWait(vw);
             }     OSWait(10);  VWRelease(vw);  return 0;  }
```

Многу значаен дел од детекцијата на топка е наоѓањето на аголот помеѓу камерата на агентот и топката. Преку примерот се гледа неговата функционалност која при симулацијата солидно се покажа како задоволителна. Пред да го пресмета аголот мора во себе да вклучува и контролор кој ќе препознава дали топката е во центарот на сликата, и со тоа да го определи што е можно попрецизно аголот. Во следниот пример со коментар е опишан соодветниот контролор.

2.2.9.3 Пресметува агол помеѓу роботот и топката

Во овој програм вршиме пресметка на аголот помеѓу роботот и топката. Тоа го правиме со две последователни движења на роботот и во зависност од поместувањето на сликата ја правиме пресметката на аголот.

```
float get_angle(float diffx, float diffy, float phi)
{
 float diffphi;
  /* агол за бараната позиција */
 if (fabs(diffx) < epsilon && fabs(diffy) < epsilon)
diffphi = phi;
 /* точката на потрага  -> нема промена на аголот */
else
   diffphi = DiscAtan((int)(1000.0 * diffy), (int) (1000.0 * diffx));
  /* промена при следното придвижување */
 diffphi -= phi;
  /* аголот е секогаш помеѓу 0 и 2*PI */
 if (diffphi >= 2.0 * PI)
   diffphi -= 2.0 * PI;
 if (diffphi <= 0)
   diffphi += 2.0 * PI;
  /* аголот е секогаш помеѓу  -PI и PI*/
 if (diffphi >= PI)
   diffphi -= 2.0 * PI;
  return diffphi; }
```



Слика 18. Детекциски агол и домет на секој од сензорите

Figure 18. Detection angle and range of everyone sensors

На слика со број 18 е прикажана скица на фудбалскиот агент EyeBot со приказ на видното поле на камерата и соодветниот распоред на инфрацрвените сензори и нивната позициона област. Тоа е од особена важност, бидејќи правилниот распоред на сензорите и нивната помала бројност се услов за динамичен систем кој ќе има голема брзина на реакција и мало време на процесирање на сигналите добиени од сензорите.

Пример во кој се опфатени главните операции на контролорите за фудбалските агенти EyeBot.

При активирање на програмот најпрво се проверува дали копчето KEY3 е притиснато и поставува знаменца соодветно на него.

```
void PPend_test();
/**Потоа прикажува грешка и става крај на програмот */
void error(char *str);
/** Програмот ќе ни врати минус еден за влез помал од нула, и единица за влез поголем од нула и нула за влез еднаков на нула */
```

```
float fsign(float number);
/** Кажи кој играч е голман */
int I_am_goalie();
/** постави параметри од тастатурата. */
float set_fparam(char text[], float minp, float start, float maxp,
float inc);
/** постави целобројни параметри од тастатурата. */
int set_iparam(char text[], int minp, int start, int maxp, int inc);
/** функција за пресметка на аркус тангенс користејќи само целобројни променливи */
float DiscAtan(int dy, int dx);
```

Во следниот пример се врши иницијализација на агентот и читање на сензорите.

```
 int InitPSD();
/** Тред за проверка на предниот инфрацрвен сензор и виртуелниот амортизер кој е со прекинувач
/** редоследно поставување на знаменцата и нивно тестирање.
void PPobstacle_test();
```

Во следниот пример се иницијализира сликата од камерата и се анализира истата.

```
void InitCam();
/** Промена на осветленоста, контрастот и обоеноста која во овој пример е по најпрво подесување.
void set_cam_parameters();
void set_img_parameters();
/** Поставување на бојата на топката кога е во центарот на сликата.  */
void set_ball_colour(colimage img);
/** Поставување на бојата на голот кога е во центарот на сликата */
void set_goal_colour(colimage img);
/** Конверзија на сликата од боја во црно бела заради прикажување на дисплејот.
   и маркирање на бојата на топката */
void mark_object(image greyimg, int x_middle, int y_middle,
int object_size);
```

/** Барање на региони во редовите, се мисли на вредност на бојата на топката во централниот регион, и враќање на координатите на центарот */
int find_ball(colimage img, int *x, int *y, int *size);
/** Барање на региони во редовите, се мисли на вредноста на голот во централниот регион, и враќање на координатите од центарот на голот */
int find_goal(int *x, int *y, int *size);
/** Тред за анализа на сликата кој се однесува на тоа да ја најде топката и да ја провери позицијата на камерата со поставување на знаменце на видена топка и знаменце на прифатена топка . */
void PPball_test();
/** Ја зема позицијата на топката во полето кога е надвор од видниот агол на топката и преземање на рутина за ориентација до непријателскиот гол */
void get_ball_coord(PositionType *ball);
/** Ја зема тековната вредност за бојата на топката */
int get_ball_col();
/** Ја зема тековната вредност за бојата на голот */
int get_goal_col(int *blue_goal);
/** Промена на RBG во HSV – со користење само на обоеноста */
int RGBtoHue(BYTE r, BYTE g, BYTE b);
Пример со кој поставуваат параметрите за движење
/** Иницијализација на VW интерфајс. */
int InitDrive();
/** Промена на позицијата за движење. */
void set_drv_parameters();
/** Ги прикажува x и y координатите во сантиметри и ориентација во степени на графичкиот дисплеј */
void print_pos();
/** Поставување на x и y позицијата. (0,0) = центар на сопствениот гол,
гледајќи напред е противничкиот гол и y оди позитивно нагоре, x е позитивно надесно,
аголот е позитивен надесно, а негативен налево */
void set_coordinates();
/** Тред за движење на голманот (GOAL KEEPER) и излезен текст во кој е даден статусот на движењето на графичкиот дисплеј.

Активирање на движењето на роботот проследено до флеговите и флегот. **/
obstacle_flag, got_ball_flag and see_ball_flag, concerning
void PPdrive_goal();
/** Тред за придвижување на агентот (FIELD PLAYER) и неговиот статус да се испише на графичкиот дисплеј.**/
obstacle_flag, got_ball_flag and see_ball_flag, concerning
void PPdrive_field();
Пример во кој се иницијализираат и поставуваат сервомоторите.
void InitServos();
/** Реализирај ги сервомоторите. */
void ReleaseServos();
/** Придвижи ја камерата нагоре и надолу и постави ја нејзината позиција.
void move_camera(int new_pos);
/** Удирање на топката (ова се реализира само од предниот дел на роботот). */
void kick_ball();

Слика 19. Прикажан е Робокупот кој е составен од пет EyeBot агенти

Figure 19. Robo Cup which is composed of five EyeBot agents

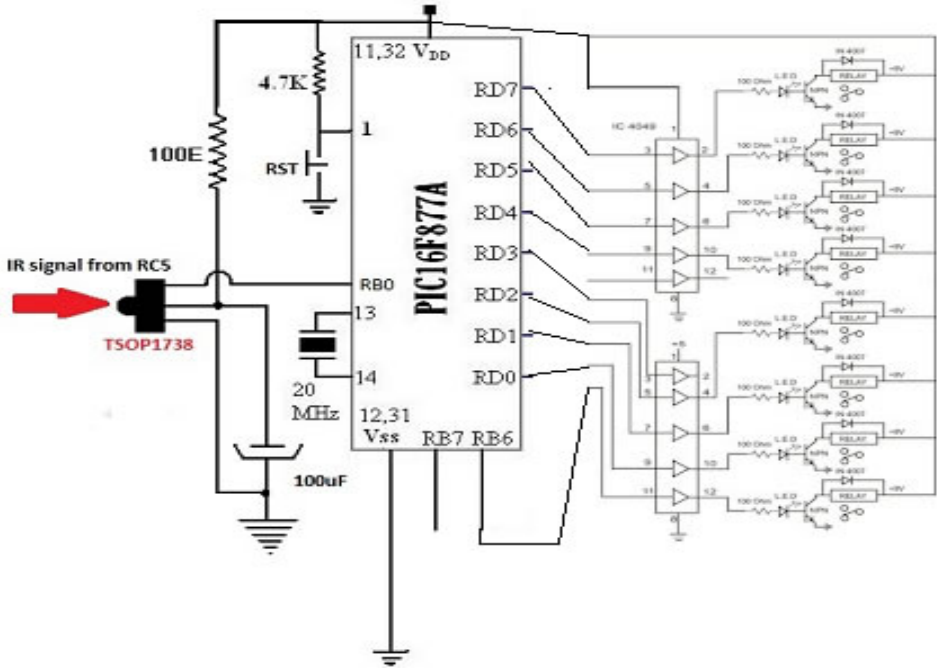
2.3 Развојни роботски платформи

Во оваа магистерска работа се опфатени и други модели на роботи кои ќе бидат подвижни и ќе имаат безжична комуникација. Врската со главниот компјутер преку кој се примаат и се праќаат соодветни наредби од клиентска страна е со посебни комуникациски модули. За илустрација може да се направи нов прототип на робот кој ќе ги прима наредбите од главниот компјутер преку инфрацрвени, ултразвучни команди во облик на командите искористени од протоколите за комуникација помеѓу уредите.

2.3.1 Робот со инфрацрвени команди

За да креираме софистициран робот кој ќе се одликува со голема брзина и со голема процесирачка моќ потребно е да одбереме брз микропроцесор кој ќе може да одговори на големиот број потреби кои ги бара фудбалскиот куп на роботи. За да ги задоволиме овие потреби препорачливо е да се искористи популарниот микроконтролор на фирмата Microchip 16F877A. Овој микроконтролор во целост може да ги задоволи потребите на телероботиката или поточно веб базираната далечинска контрола на роботи. Се одликува со 40 пина и вкупно пет порти или вкупно имаме 33 пина на располагање за праќање и примање команди на влезно-излезните уреди. За комуникација со главниот компјутер користиме два пина на PORTB. Го користиме 0 пин за примање на инфрацрвена команда и за праќање на инфрацрвена командата на 7-от пин. На овој начин можеме да управуваме огромен број на роботи и сите да ги примаат командите преку само еден влезно-излезен уред. Со ова ќе постигнеме намалување на трошоците за набавка на роботи кои се повеќе пати на цена, а ќе добиеме иста, скоро идентична функционалност. На слика со реден број 20 е прикажана електричната шема на овој робот. Командите пристигаат на пинот означен со RB0 како сериска низа од последователни нули и единици. Во овој магистерски труд изработен е програм кој се извршува на овој микроконтролер. Програмирано е декодирање на протоколот на фирмата Сони, популарниот Sony-Ir протокол. Овој робот ќе содржи три сервомотори со кои ќе се извршуваат сите команди од клиентска страна. Повратната врска за моменталните позиции на секој од роботите ќе ги дава камерата која ќе ги

снима на самиот врв високо над сите роботи. Овој модел на робот има најбрз одѕив бидејќи не е потребно да се прави конекција со управувачкиот компјутер. Тоа веќе е направено на самиот почеток кога се прави главниот програм за секој од овие роботи. Во секој од микроконтролорите се инсталира програм кој ќе содржи адреса која ќе биде уникатна за тој робот. Со тоа се постигнува следново, кога ќе праќаме команда од главниот компјутер таа е во облик на инфрацрвен сигнал. Овој сигнал во себе содржи две информации команда и адреса. Овој сигнал го добива секој од овие роботи. Секој од роботите бидејќи има сопствена адреса не постои можноста од мешање на командите.

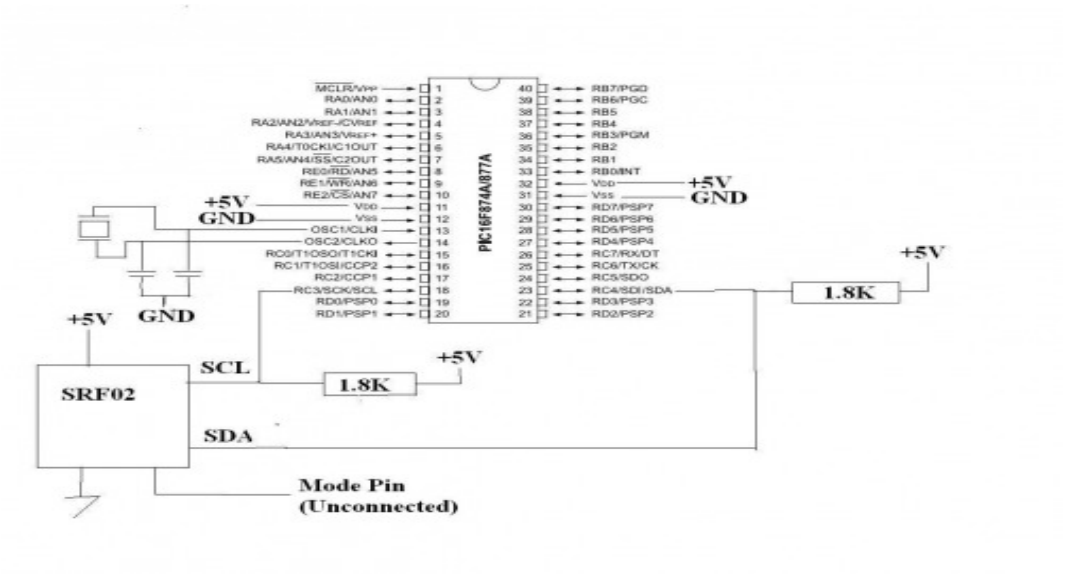


Слика 20. Прикажана е електричната шема за врската помеѓу микроконтролорот и инфрацрвениот сензор

Figure 20. Shown is the electrical connection scheme between the microcontroller and infrared sensor

2.3.2 Робот со ултразвучни команди

Во поглед на комуникацискиот модул исто така можеме да користиме и ултразвучен сензор преку кој ќе ги добиваме командите од изворот на ултразвучните давач. Овој модул ќе биде во сериска комуникација со главниот компјутер. Ултразвукот е нечуен за нас и затоа нема да смета на околината при правилно извршување на командите кои се задаваат од клиентска страна. Исто како и претходниот разработен робот и овој робот ќе ги прима сите команди од главниот компјутер. Само таа команда која ќе ја содржи точната адреса ќе се изврши на зададениот робот. Исто како и со инфрацрвените команди по иста логика и овој модул ги прима и ги праќа командите до главниот компјутер. Постои можност сами да направиме комуникациски протокол по истата логика како и за инфрацрвените протоколи, со активирање на почетен импулс кој е по должина на траење подолг од останатите вредности на нули и единици. Откога ќе престане траењето на стартниот сигнал правиме мала пауза и потоа ги пуштаме сигналите за нули и единици. Можеме слободно да направиме сигналот за единица да трае два пати подолго од сигналот за нула, а со тоа ќе ја елиминираме секоја грешка која може да настане при праќањето на командата.

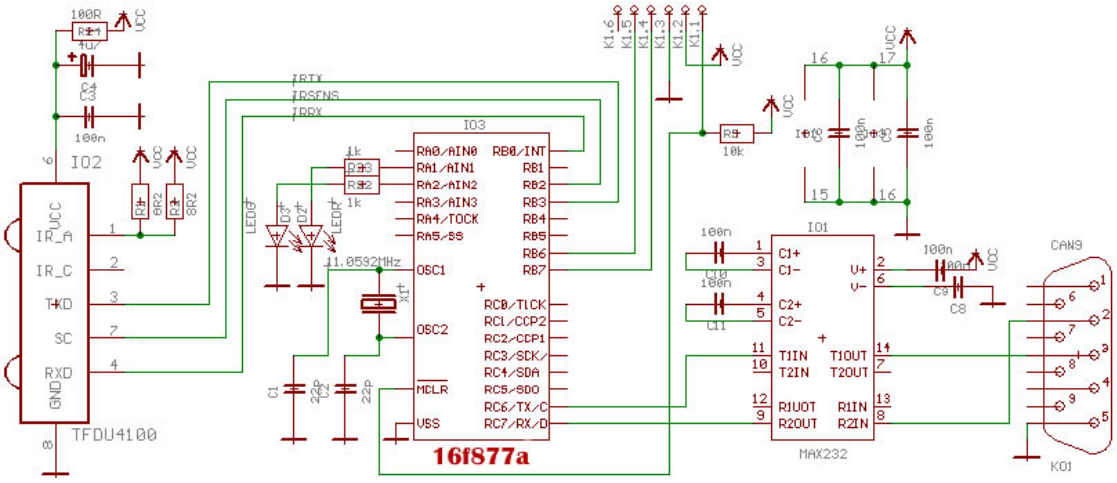


Слика 21. Прикажана е електричната шема за врската помеѓу микроконтролерот и ултразвучниот сензор

Figure 21. Show electrical connection scheme between the microcontroller and ultrasound sensors

2.3.3 Робот со IRDA комуникациски протокол

Преку овој уред се примаат и се праќаат командите до секој од роботите. Секој од роботите реагира на различна команда така што веројатноста некој од роботите да прими друга команда е нула. Овој систем нуди подобра комуникација од Bluetooth комуникацијата бидејќи нема потреба од остварување на претходна комуникација помеѓу главниот компјутер и роботите. Тоа е последица поради немањето на оптички пречки во работната околина. Инфрацрвениот давач на команди е поставен високо над секој од роботите во Z оска. Секој од роботите делува во X и Y оска и затоа овој начин на комуникација е побрз и постабилен од Bluetooth комуникацијата.



Слика 22. Прикажана е електричната шема за врската помеѓу микроконтролерот серискиот порт на главниот управувачки компјутер и инфрацрвениот влезно излезен уред

Figure 22. Show electrical connection scheme between the microcontroller serial port on the main control computer and infrared input output device

Програмскиот код за декодирање на sony-ir протоколот е имплементиран во микроконтролорот PIC16F877A. Програмирано е во програмскиот јазик C++ и се компајлира во програмот MPLAB кој е најупотребуваниот програм за програмирање на микроконтролори. Во пример под реден број 2, е прикажан

кода за декодирање на Sony-Ir протоколот кој се користи за микроконтролорот PIC16F877A.

Пример 2. Декодирање на Sony-Ir протокол

```
void FCM_decode()
{
	//Delay
	//Delay: 3 ms // софтверско доцнење од 3 милисекунди за стартниот импулс кој трае 2,4 милисекунди и по него со пауза од 0,6  милисекунди
	delay_ms(3);
	//Input
	//Input: B0 -> var_decode
	trisb = trisb | 0x01;
	FCV_VAR_DECODE = ((portb & 0x01) == 0x01);
	//Output
	//Output: var_decode -> D0
	trisd = trisd & 0xFE;
	if ((FCV_VAR_DECODE))
		portd = (portd & 0xFE) | 0x01;
	else
		portd = portd & 0xFE;
	//Delay
	//Delay: 600 us
	delay_us(255);
	delay_us(255);
	delay_us(90);
	//Input
	//Input: B0 -> var_decode
	trisb = trisb | 0x01;
	FCV_VAR_DECODE = ((portb & 0x01) == 0x01);
	//Output
	//Output: var_decode -> D1
```

```
trisd = trisd & 0xFD;
if ((FCV_VAR_DECODE))
	portd = (portd & 0xFD) | 0x02;
else
	portd = portd & 0xFD;
//Delay
//Delay: 600 us
delay_us(255);
delay_us(255);
delay_us(90);
//Input
//Input: B0 -> var_decode
trisb = trisb | 0x01;
FCV_VAR_DECODE = ((portb & 0x01) == 0x01);
//Output
//Output: var_decode -> D2
trisd = trisd & 0xFB;
if ((FCV_VAR_DECODE))
	portd = (portd & 0xFB) | 0x04;
else
	portd = portd & 0xFB;
//каснење
// каснење: 600 us
delay_us(255);
delay_us(255);
delay_us(90);
//Input
//Input: B0 -> var_decode
trisb = trisb | 0x01;
FCV_VAR_DECODE = ((portb & 0x01) == 0x01);
//Output
//Output: var_decode -> D3
```

```
	trisd = trisd & 0xF7;
	if ((FCV_VAR_DECODE))
		portd = (portd & 0xF7) | 0x08;
	else
		portd = portd & 0xF7;
	//Delay
	//Delay: 600 us
	delay_us(255);
	delay_us(255);
	delay_us(90);
}
void main()
{
	//Initialization
	adcon1 = 0x07;
	//Interrupt initialization code
	option_reg = 0xC0;
	//Loop
	//Loop: While 1
	while (1)
	{
		//Interrupt
		//Interrupt: Enable RB0INT
		st_bit(option_reg,INTEDG);
		st_bit(intcon,GIE);
		st_bit(intcon, INTE);
	}
	mainendloop: goto mainendloop;
}
// Макро за ново настанат интерапт
void MX_INTERRUPT_MACRO(void)
{
```

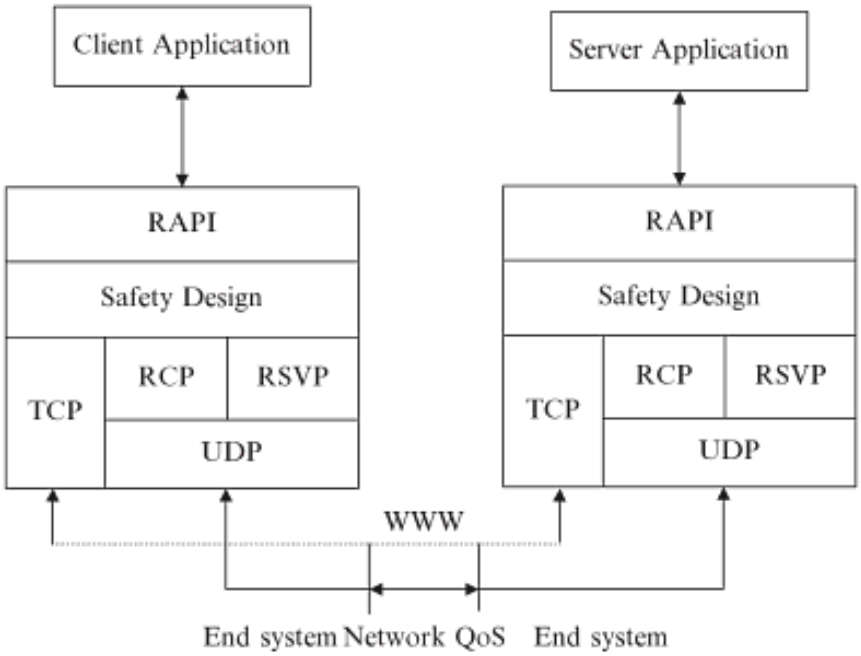
```
//Handler code for [RB0INT]
#ifndef MX_INTHANDLER_intcon_INTF
#define MX_INTHANDLER_intcon_INTF
if (ts_bit(intcon, INTF) && ts_bit(intcon, INTE))
{
        #ifdef USE_FLOWCODE_ICD
        extern char ICD_Interrupt_Enable = 1;
        #endif
        FCM_decode();
        cr_bit(intcon, INTF);
        #ifdef USE_FLOWCODE_ICD
                extern char ICD_Interrupt_Enable = 0;
        #endif
} #else #warning "This interrupt has previously been enabled, so the macro
<decode> may never get called." #endif
}
```

3. ЦЕЛ НА ИСТРАЖУВАЊЕТО

Со развојот на новите технологии роботиката полека станува наша потреба и секојдневие. Како пример: може да се искористи за управување на далечина во својот дом на живеење или место на работа. Целта на ова истражување е да се направи труд кој ќе биде широко употреблив. Потребата од имплементација на готови софтверски решенија за контрола на роботите на далечина е голема. За таа цел потребно е од постојните комуникациски протоколи да одбереме кој е најсоодветен за управување на роботите на далечина. Често се користени трите протоколи:

- CGI Common Gateway Interface
- TCP / IP Transport Control Protocol/Internet Protocol
- UDP / IP Universal Datagram Protocol/Internet Protocol

Го користиме UDP / IP протоколот бидејќи е побрз од останатите т.е. потребно е помалку време во праќањето на командата од серверска на клиентска страна. Во споредба со останатите два протокола има помала сигурност во однос на пратените пакети во споредба со другите два кои нудат поголема сигурност во тој поглед. За управувањето на роботите на далечина брзината на извршување на командите е од особена важност. Затоа најчесто се користи UDP/IP протокол. Доколку користиме комбинација од двата протокола TCP/IP и UDP/IP ќе имаме комбинирана архитектура. На слика под реден број 23, е прикажана архитектурата со комбинирани протоколи која ги задоволува потребите во управувањето на роботи на далечина. Методите кои ќе се користат се сигурни и нивното активирање и извршување зависат само од интернет провајдерите кои нудат услуги за стабилен и брз интернет. За да најде широка примена трудот ќе се користи мрежниот протокол HTTP и неговиот метод POST кој е клучен во активирањето на наредбите од серверска страна. Денеска постојат многу мрежни протоколи кои овозможуваат стабилност при преносот на информациите. Основен проблем во управување на далечина е брзината на одзив на системот. Временското доцнење е најголемата пречка за овие системи. За да го намалиме ова временско доцнење кое за потребите на робокупот е од особена важност го користиме претходно спомнатиот HTTP протокол. Перформансите на комуникациската мрежа се карактеризирани со следниот QualityOfServices (QOS) модел во кој се инволвирани четири параметри. Првиот параметар е временското доцнење кое го бележиме со „T“. Вториот параметар е мрежна напнатост која ја бележиме со „J“. Третиот параметар е протокот на информации кој претставува податочно трансмисиона рата. Четвртиот параметар е губењето на пакетите. Во мрежата во зависност од уредите кои се користат се случува да се изгубат пакетите за да преку методи за детекција повторно изгубените пакети се препраќаат. Со ова се постигнува стабилност и сигурност во комуникацијата.



Слика 23. Архитектура со комбинирани протоколи која ги задоволува потребите во управувањето на роботи на далечина

Figure 23. Architecture with combined protocols who meet the needs in the management of remote robots

• RCP: Real Control Protocol (реален контролен протокол).
• RAPI: Real-Time Application Programming Interface (апликациско-програмски интерфејс во реално време).
• SDL: Safety Design Layer (безбедносен слој).
• RSVP: Resource Reservation Protocol (ресурсно-резервационен протокол).

3.1 HTTP протокол

Протоколот за пренос на хипертекст (англиски: Hypertext Transfer Protocol, скр. HTTP) е мрежен протокол. Развојот е започнат во 1989 година на CORN како резултат на потребата од еден текстуален линк со некоја содржина да може да се отвора на повеќе компјутери креирани од различни научници. Резултатот од таа работа е концептот на „hypertext“ кој дозволува информацијата да биде во веб структура. На информационите нодови може да им се постави линк до останатите нодови на повеќе начини. Со ова им се дозволува на корисниците динамичко повикување на информацијата. HTTP е основа за комуникација на World Wide Web. Создаден е како средство за објавување на HTML страници. Развивањето на стандардот е координирано од (IETF) Internet Engineering Task Force и World Wide Web Consortium. HTTP е протокол за комуникација помеѓу опслужувачот и клиентот, кој функционира на принцип барање одговор. HTTP клиентот, кој обично е веб-прелистувач, иницира пренос на податоци по креирањето на TCP / IP врска со оддалечен опслужувач на одредена порта. Клиентот го поднесува HTTP барањето на опслужувачот. Опслужувачот кој содржи податоци, обезбедува ресурси, како што се HTML датотеки или врши други работи во име на клиентот, и на крај враќа одговор порака на клиентот, а одговорот на проектот содржи информации за статусот на барањето и може да содржи барања за клиентот. HTTP бил измислен од страна на Тим Бернерс-Ли. Во тоа време, File Transfer Protocol (FTP) е веќе достапен за пренос на датотеки. Оваа верзија на HTTP поддржува виртуелни сервери, кеш за управување и за идентификација. За потребите на оваа магистерска работа користиме основен HTTP преку кој користиме пост метода за праќање и примање на локалните променливи. HTTP POST методата дозволува формите да бидат испратени на дискретен начин. За да не ги испраќа вредностите како дел од URL, вредностите се праќаат невидливо како дел од телото на HTTP барањето. Механизмот позади сето ова сега засега не е важен, важно е дека ова ни помага да можеме да испратиме променливи кои се цело време скриени на начин со кој истите нема да бидат покажани во URL на барањето.

4. МЕТОДИ НА ИСТРАЖУВАЧКАТА РАБОТА

Многу важен метод кој е користен во анализата и интерпретацијата е креирањето на базната линија. Базната линија служи за споредба на податоците добиени по експериментален пат. Доколку креираниот облик кој програмот се труди да го препознае не е соодветно изработен, нормално, програмот нема да може да го препознае правилно. Решението кое се нуди е да одбереме секогаш боја за примерок која најчесто не се појавува во околината, на тој начин ќе ја избегнеме секоја грешка која може да настане при процесот на препознавање на соодветната боја. Во примерот со четирите „lego nxt“ роботи врз секој од роботите е поставена коцка со соодветна боја, која најчесто ја нема во околината. Подот е фарбан со светлозелена боја додека краевите на игралиштето се со крем боја. Ова е задоволително решение иако коцката на роботот е со бела боја сепак најважно е да се одберат бои кои најчесто не се појавуваат во околината. Доколку околината е постојано променлива за да може правилно да ја одбереме соодветната боја може да се послужиме со табела во која се дадени вредностите за секоја од боите посебно и со нивен микс може точно да одбереме боја која ќе може правилно да се препознава од програмот. Со комбинација на секои од овие бои можеме да направиме боја која ќе биде уникатна и програмот ќе може за најкратко време да ја детектира и соодветно да ја следи. Ова е од особена важност бидејќи времето за правилна детекција е од највисок приоритет. Модулот кој ќе го користиме е „задршка на боја“. Овој модул се користи за да се отргнат делови од сликата кои не се во специфичен ранг на боја. Овој модул може да се користи за детекција на објекти со конзистентна вредност на бојата. Интерфејсот ја прикажува бојата во црвен, зелен и син хистограм. Хистограмот ја прикажува вредноста на пикселот (0-255) по X оска со број на пикселите (0-големина на слика) и по У оска вредноста на бојата. Користејќи го хистограмот може да ги филтрираме пикселите кои ја немаат вредноста со саканиот објект кој се следи. За да го појасниме алгоритамот поставуваме три услови за секоја од боите посебно. Доколку не е во дадениот опсег тогаш вредноста е нула.

Табела 1. Вредности на сјајност на секоја од мастило боите посебно

Table 1. Brightness values of each ink color

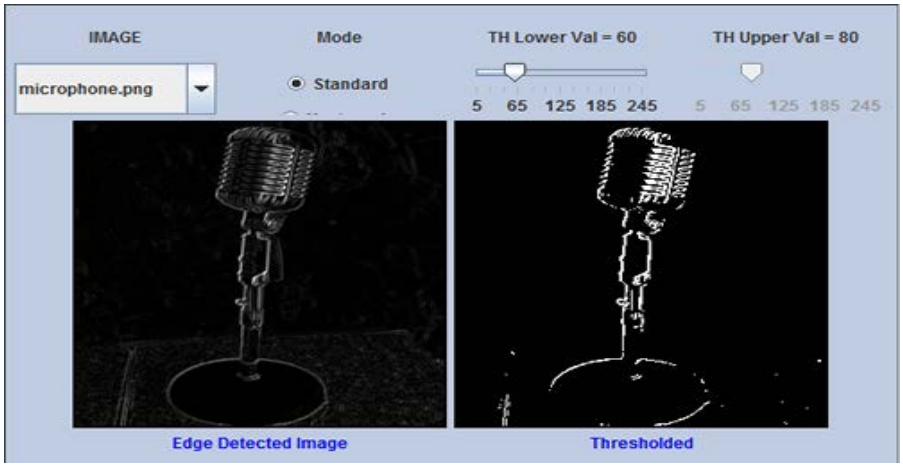
Боја на мастилото /ink color	Цијан мастило / Cyan ink		Маѓента мастило / Magenta ink		Жолто мастило / Yellow ink	
Број на капки / Number of drops	Ниво на сјајност / Level of glossiness	Разлика / Difference	Ниво на сјајност / Level of glossiness	Разлика/ Difference	Ниво на сјајност / Level of glossiness	Разлика
Стандард Standard	59.05	-35.90	49.59	-44.28	87.39	-6.65
1	69.08	10.02	60.94	11.35	87.69	0.30
2	66.83	7.78	57.01	7.42	87.44	0.05
3	67.80	8.75	55.07	5.48	87.42	0.03
4	66.86	7.81	53.81	4.22	87.37	-0.01
5	60.59	1.53	52.96	3.36	87.12	-0.27
6	58.39	0.66	50.01	-0.42	87.65	-026
7	*62*.34	3.29	53.43	3.84	87.65	2 21
8	60.17	1.12	50.49	0.90	85.53	-1.86
9	60.69	1.64	52.98	3.38	84.92	-2.47
10	61.37	2.32	48.35	-l.25	87.78	0.39

1) Ако R<R_min_задршка или R>R_max_ задршка Тогаш R=0

2) Ако G<G_min_ задршка или G>G_max_ задршка Тогаш G=0

3) Ако B<B_min_ задршка или B>B_max_ задршка Тогаш B=0

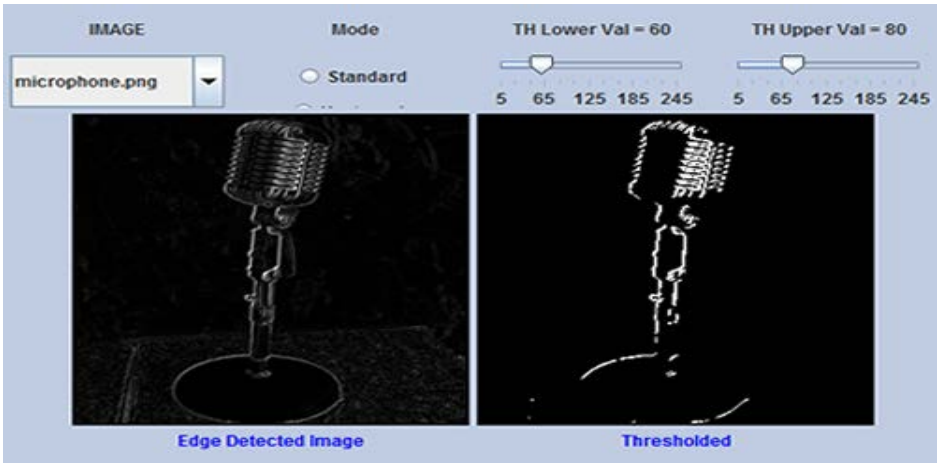
Хистограмот е подесен да го игнорира секој бел или црн пиксел во сликата. Со ова се овозможува темните или светли слики успешно да се појавуваат во хистограмот. Овие вредности може да се додадат во позадина, но тие нема да имаат ефект на операцијата при задршка на боја. Треба да забележиме дека условот за осветленост ја менува правилната вредност која му доаѓа на тековниот сегмент од сликата. Треба да специфицираме широка вредност за задршка на боја за да може да ги следиме движењата на објектот исправно.



Слика 24. Резултат од примена на модулот „задршка на боја“

Figure 24. Result of the application module „threshold“

Без разлика каков објект ќе следиме ако правилно ги одбереме границите за задршка на боја ќе направиме успешно следење на објектот. При детекција на објектот од камерата настанува промена на вредноста на секој од пикселите и сликата не е статична. Без разлика дали има или нема движење се продуцира динамична слика бидејќи драјверот кој го користи камерата користи филтри и кодеци кои прават апроксимација и оптимизација. Сликата од камерата постојано е различна исто така и поради шумовите кои се јавуваат во просторијата каде е поставена. За да се избегне ова користиме модул „хистерезис“ и постепено го зголемуваме опсегот во делот што останал при задршката на боја. Така го изолираме саканиот дел. Постапката се повторува итеративно сè додека вредноста е поголема од зададениот лимит. Се прави задршка на боја со трите услови и плус во итерација се повторува циклусот сè додека не се постигне вредноста која не е помала од зададениот лимит.



Слика 25. Резултат од примена на модулот „задршка на боја“ со активирање на модулот „хистерезис“

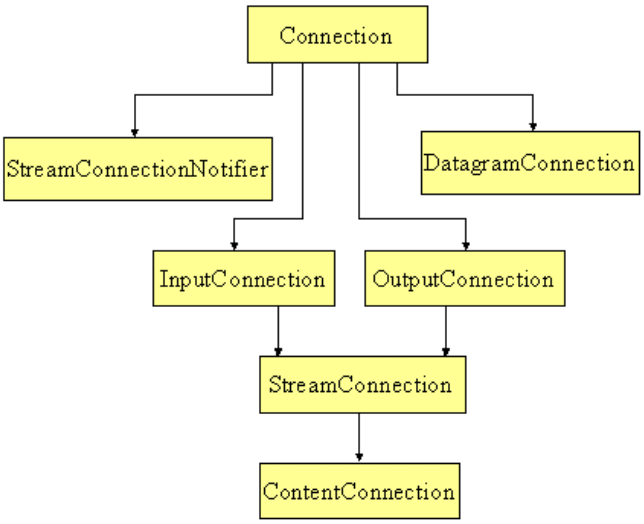
Figure 25. Result of the application module "threshold" with combination by module "hysteresis"

Можеме да уочиме дека со примената на модулот „хистерезис“ објектот е комплетно детектиран. Позадината е исчистена и само доминира активниот објект. Ова го анализираме сè со цел да најдеме решение кои ќе може сликата од камерата правилно да ја процесира и да ги следи роботите беспрекорно во какви било услови. За таа цел во овој труд ќе опишеме повеќе типови на роботи кои ќе може да се управуваат на далечина и кои ќе може со овој модул успешно да се следат. За да може да се управуваат на далечина потребно е да им инсталираме камера поставена во агол кој верно ќе ги детектира. Камерата треба да е со солиден квалитет, да има голема резолуција и да се пренесува на серверска страна за да може детекцијата и приказот на дејството да се прикажат верно на корисникот. Потоа треба да вградиме уред кој ќе ја прима командата од серверска страна и за најкратко време ќе ја извршува на роботот. Уредите кои се користат во овој труд се инфрацрвен приемник, безжичен Bluetooth уред, безжичен wi-fi модул, ултразвучен модул, GMS модул и IRDA модул. За секој од роботите кои се цел на истражување даваме детален опис на можностите кои ги нуди. Во дел од нив се дадени и кратки програми за иницијализација и тестирање на истите. Секој од применетите софтвери во овој труд се бесплатни и слободни за користење во научно-истражувачки работи. Основен дел во овој труд е препознавање на местоположбата на секој од роботите и затоа користиме повеќе методи за препознавање на облиците.

4.1 Модул за веб далечинска контрола на роботи

Во овој магистерски труд изработени се два модула, едниот клиентски, а другиот серверски. Клиентскиот програм претставува една флеш апликација преку која се следат сите роботи со можност за нивна контрола. Програмите кои служат за контрола на роботите преку интернет се legoonline1.exe, legoonline2.exe, legoonline3.exe, legoonline3.exe, robotread.swf, robotdetection.swf, game.sql, insert1.php, insert2.php, insert3.php, insert4.php, insert.php, read.php, reset.php, robotdetection.html, robotread.html. Секој од овие модули е детално објаснет во оваа магистерска работа сѐ со цел да може да се направи надградба на постојната верзија и да може овој магистерски труд да биде во широка употреба кај поголемиот дел од програмерите и дизајнерите на роботи ширум светот и да трпи понатамошни надградби. Модулот „legoonline“ служи за контрола на роботот преку HTTP протокол. Во себе вклучува класи со кои се остварува врска со HTTP протоколот и Bluetooth комуникација. Целта на овој модул е да покаже на кој начин може да управуваме робот преку интернет. За таа цел го користиме роботот на lego nxt2 кој преставува најприменуван робот во пракса кај широката јавност. Останатото што ни е потребно е уред Bluetooth преку кој ќе се извршуваат сите протоколи и настани без притоа да бара модулот да се активира рачно настанот и да не прашува за дозвола. За да направиме успешен модул го користиме најновиот програм dev-cpp. Креираме проект во него и притоа дефинираме главен програм. Пред да почнеме да го објаснуваме програмскиот код да кажеме нешто и за начинот преку кој ќе работи целиот систем. Преку форма во флеш креираме копчиња преку кои повикуваме преку php скрипта “sendandload” метод преку која ги праќаме варијаблите од флеш апликацијата. За тоа време програмот постојано прима податоци од веб-страната, а програмот постојано чита што има во текст фајлот како содржина. Во зависност од содржината на текст фајлот се дава соодветната команда на роботот преку претходно овозможена Bluetooth конекција. Ваквиот начин на работење може да овозможи широка примена насекаде во техниката. Како пример може да се искористи за вклучување на некоја, на некое реле, мотор и.т.н. Овој програм може да се искористи во фудбалски куп на роботи. Преку претходно

изработена веб-страна се регистрира секој од играчите. Потоа има пристап во управување на соодветно одбраниот робот. Мешање на командите не е возможно бидејќи претходно со парнување помеѓу компјутерот и роботот се избегнуваат секакви грешки во конекцијата. Зашто секој од роботите работи на посебен порт и адреса. Се исклучува секакво мешање на командите на секој од роботите. На сликата се прикажани двајца ученици како играат фудбал помеѓу себе со овој програм локално. Исто може да се прошири на голем број на играчи на тој начин можеме да реализираме цел еден фудбалски тим. Играње на фудбал се прикажува на веб страната и играчите да ја играат играта онлајн. Исто така може да се управува и да се следи и на своите мобилни телефони доколку имаат инсталирано флеш апликација соодветен за својот телефон. Целта на овој програм е да има широка примена и да го зголеми интересот за технологијата. Друга примена може да најдеме во надзор на некој систем, пример: подвижен робот кој чува некој објект и преку мобилниот телефон со отворање на страната ќе прави праќање на соодветната команда до самиот робот. На тој начин ќе реагираме соодветно на саканиот настан во реално време. За да го разбереме подобро целиот систем како функционира ќе го објасниме конекцискиот интерфејс. На слика под реден број 26 е прикажана хиерархијата на конекцискиот интерфејс, каде се гледаат врските и насоката помеѓу конекцијата и содржинската конекција. Во фрејмворкот CLDC Generic Connection сите конекции се креирани користејќи отворен статички метод од класата “Connector”. Ако е успешен овој метод враќа објект кој ќе имплементира еден од конекциските интерфејси. Овој конекциски интерфејс е базен интерфејс.



Слика 26. Хиерархијата на конекцискиот интерфејс

Figure 26. Hierarchy of connection interface

4.2 Bluetooth конекција

Овој програм последователно прави две конекции: една со веб страната и другата е Bluetooth конекција со роботот. Bluetooth технологијата е искористена во овој робот исто така и во самиот програм. Потребата од тоа е од особена важност бидејќи со неа се зголемува слободата на движење, а тоа поточно значи дека немаме потреба од кабел туку се остварува безжично. Примена наоѓа во пренос на фајлови, AD-HOC мрежа преку која спонтано може да формираат заедница од мрежи со произволно времетраење. Друга примена наоѓа во синхронизација на уредите преку конекции со други преносни компјутери, pda, компјутери и мобилни телефони кои им дозволуваат да ги апдејтуваат информациите на повеќе други уреди автоматски доколку некој уред направи промени. Друга примена е користење на периферна конекција, на слобода на користење на рацете на корисникот преку безжични слушалки и микрофон, преку плаќање со остварување на врска со други апарати, пример апарат за сокови и кафе каде може корисникот со овој програм да оствари врска со уредот кој има Bluetooth уред и на тој начин преку намалување на сметката од мобилниот телефон да го плати производот и да го добие соодветно од самиот уред.

Слика 27. Примена на програмот Lego Nxt Http во основно училиште

Figure 27. Application of Lego Nxt Http into primary school

Програмот се извршува во c++ преку претходно креирање на проект и креирање на сите претходно спомнати класи. На сликата под реден број 27, е прикажано како тоа функционира во пракса. Со поврзување на три мотори A,B,C се гледа дејството. Веб-страната преку flash + php ја запишува во Mysql базата соодветната вредност за секоја од командите. Програмот е успешно тестиран и нема никакви исклучоци и падови, направени се повеќе теста на пробни управувања. Роботот има минимален број на елементи и е едноставен за составување. Инсталацијата на програмот во компјутер е брза бидејќи само се симнува програмот од веб-страницата. Пред да го активираме програмот треба да го активираме Bluetooth уредот. Откога ќе се активира го пуштаме програмот и правиме конекција со постојниот лего робот. После тоа програмот е подготвен за примање на командите од веб-страната. Програмот е поставен на следната интернет адреса http://www.robotsonline.info За да работи правилно потребно е веб пребарувачот да има инсталиран флеш апликација која доколку ја нема корисникот автоматски ќе се појави коментар за одобрување на негово инсталирање. На полето десно се запишуваат извршените команди и добиваме статус на соодветните команди.

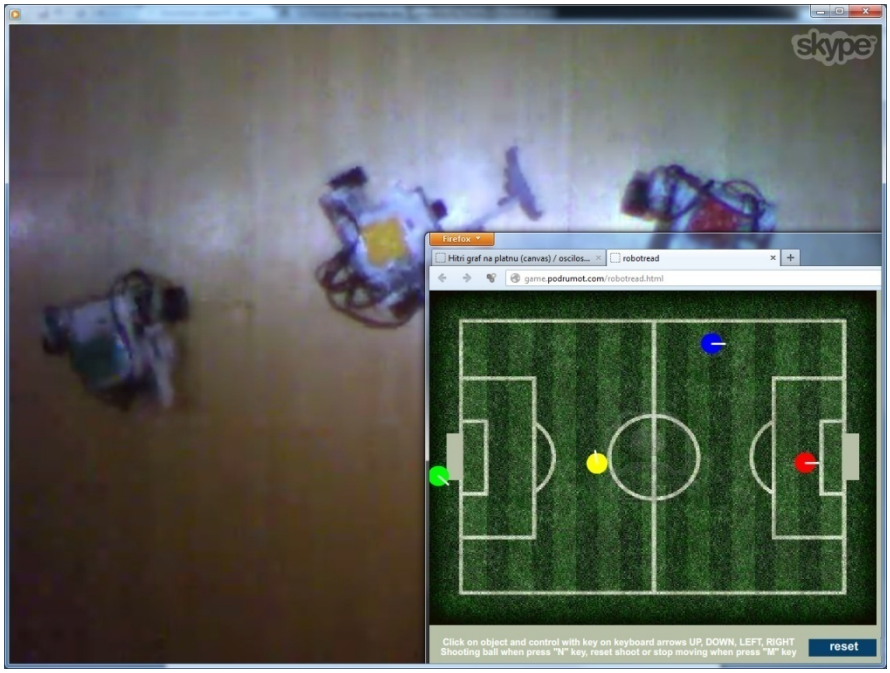
4.3 Главен програм за следење на роботите

Програмот „robotdetection.swf“ е програмиран во adobe flash cs6 во лиценцирана школска верзија. Програмот во себе вклучува пет скрипти Color32.as, CustomVideo.as, DragObject.as, BmpGenerator.as, ColorTracker.as и главната програмска скрипта Main.as во која се повикуваат сите функции кои се потребни за правилна детекција на роботите. За успешно да се разликува еден робот од друг секој од четирите роботи носи коцка со различна боја врз него. Првиот робот има црвена коцка, вториот има сина коцка, третиот робот има зелена коцка и четвртиот робот има жолта коцка. Коцките се прицврстени на роботот и не сметаат на движењата на роботите. Во скриптата ColorTracker.as е користена наредбата „threshold“ преку која правиме филтер на претходно одбраната боја од самиот визуелен приказ на екранот. Се поставуваат граници во кои да се прави толеранцијата на претходно избраната боја за да може да се намалат шумовите кои настануваат кога се движи објектот. Исто така се користи и наредбата getPixel32(loc2, loc1) која ни ја дава вредноста на пикселот кој е на loc2 и loc1 соодветно по x и y оска во претходно иницијализираните димензии на видеото кое го правиме со наредбата CustomVideo(640,480). Со оваа наредба дефинираме во кои димензии да биде сликата од видеоизворот, во нашиот случај веб камерата.

Пример за тоа како се користи оваа наредба на сите четири роботи.

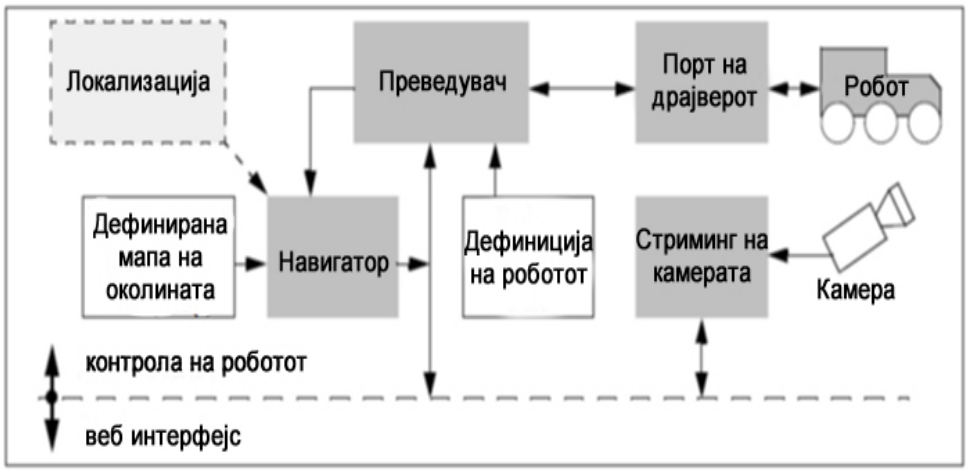
```
private var _searchRange:int=25; // ранг на пребарување е поставен на вредност 25
while (loc1 < _position.y + _searchRange) // значи правиме анализа на сликата од камерата од пиксел до пиксел на растојание од 25 пиксели за да може брзо да го прати саканиот робот кој претходно е избран од сликата.
        {
            loc2 = _position.x - _searchRange;
            while (loc2 < _position.x + _searchRange)
            {
```

```
            if (_trackBmd.getPixel32(loc2, loc1))      // ja земаме вредност од соодветниот пиксел доколку не постои таква вредност не ја зголемуваме сумата по икс и ипсилон оска
            {
                _xSum = _xSum + loc2;
                _ySum = _ySum + loc1;
                var loc3:*;
                _xCnt++;
                _yCnt++;
            }
            ++loc2;
        }
        ++loc1;
    }
_trackBmd.threshold(_trackBmd, _trackBmd.rect, new Point(), "<", _r1.color, _fillClr, 4294901760);  // ги дефинираме границите во кои ќе важи условот
        _trackBmd.threshold(_trackBmd, _trackBmd.rect, new Point(), ">", _r2.color, _fillClr, 4294901760);  // ги дефинираме границите во кои ќе важи условот

        _trackBmd.threshold(_trackBmd, _trackBmd.rect, new Point(), "<", _g1.color, _fillClr, 4278255360); // ги дефинираме границите во кои ќе важи условот

        _trackBmd.threshold(_trackBmd, _trackBmd.rect, new Point(), ">", _g2.color, _fillClr, 4278255360); // ги дефинираме границите во кои ќе важи условот

        _trackBmd.threshold(_trackBmd, _trackBmd.rect, new Point(), "<", _b1.color, _fillClr, 4278190335); // ги дефинираме границите во кои ќе важи условот

        _trackBmd.threshold(_trackBmd, _trackBmd.rect, new Point(), ">", _b2.color, _fillClr, 4278190335); // ги дефинираме границите во кои ќе важи условот
```



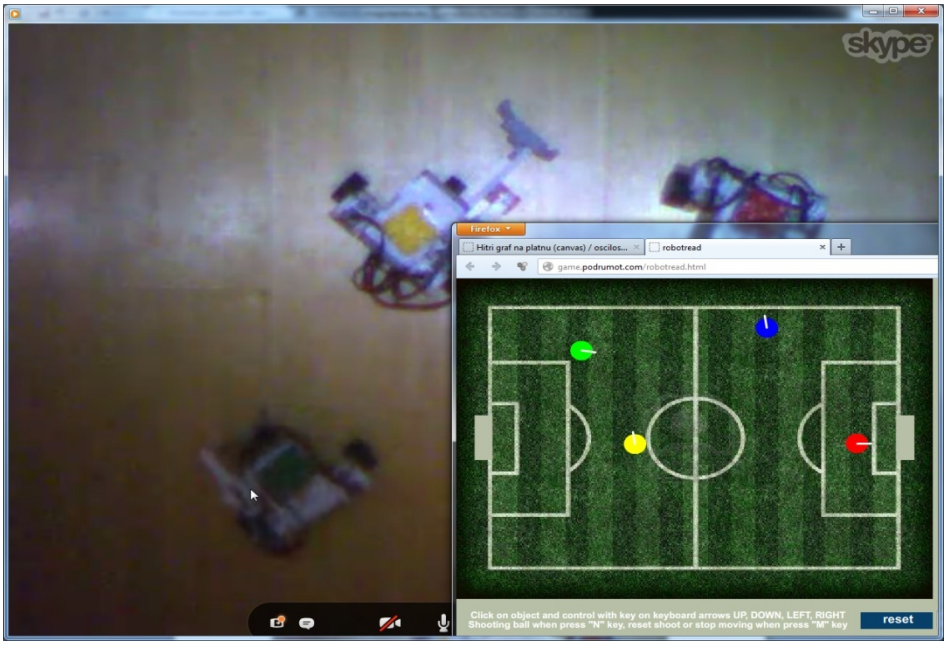
Слика 28. Форма за активирање на роботите и активирање на командите и пренос на слика од реалните роботи преку видеостримингот на сервисот скајп

Figure 28. Form to activate the robots and activation commands with image transfer by actual robots through video streaming service on Skype



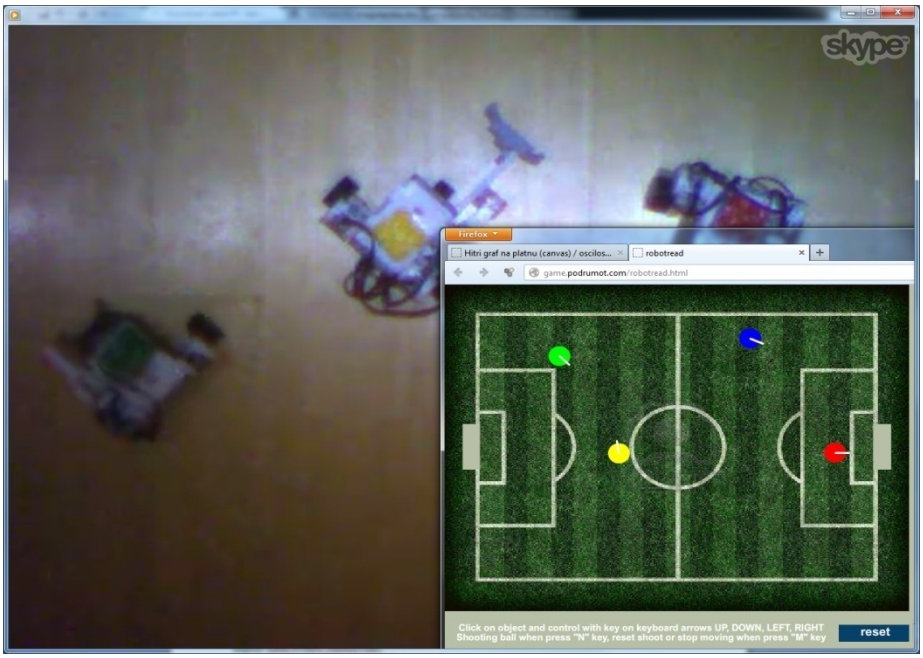
Слика 29. Блок дијаграм преку кој се прикажува текот на информацијата од камерата до моментално управуваниот робот

Figure 29. Block diagram through which demonstrates the information from the camera to the currently controlled robot



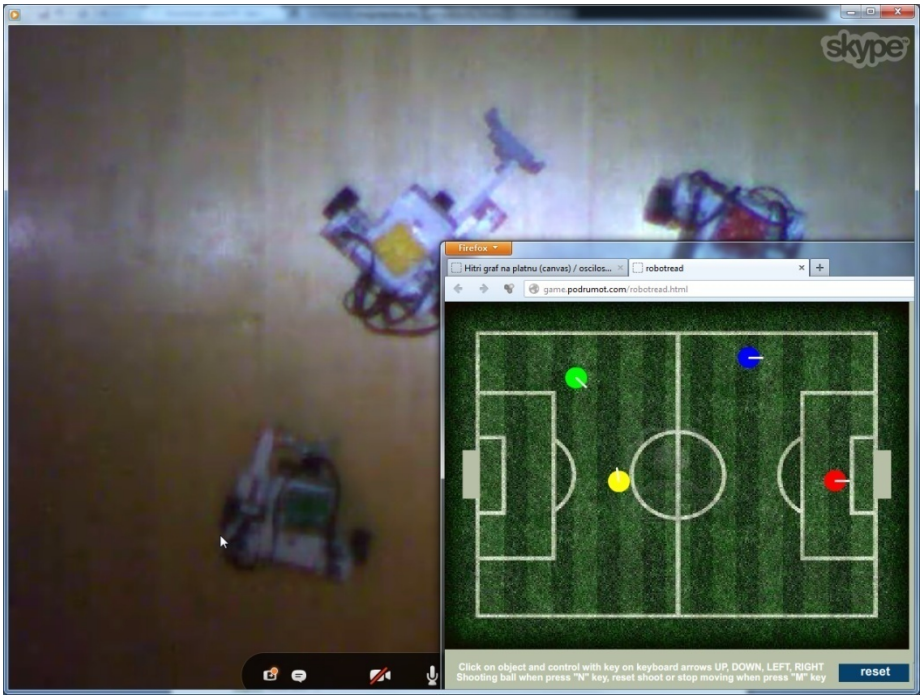
Слика 30. Форма за активирање на роботите и активирање на командите и пренос на слика од реалните роботи преку видеостримингот на сервисот скајп

Figure 30. Form to activate the robots and activate commands and transfer an image of actual robots through video streaming service Skype



Слика 31. Форма за активирање на роботите и активирање на командите и пренос на слика од реалните роботи преку видеостримингот на сервисот скајп

Figure 31. Form to activate the robots and activation commands and transfer of pictures from the actual robots through video streaming service Skype



Слика 32. Форма за активирање на роботите и активирање на командите и пренос на слика од реалните роботи преку видеостримингот на сервисот скајп

Figure 32. Form to activate the robots and activation commands and transfer an image of actual robots through video streaming service Skype

Во главниот програм Main.as ги активираме сите функции потребни за правилна детекција на секој од роботите поединечно. Во овој програм селекција на саканиот робот правиме со кликнување на некое од обоените кругови. Во нашиот пример обоени се со сина, зелена, црвена и жолта боја. По четири последователни клика на прозорецот од програмот се активира детекцијата на објектите по боја. Потоа настанува следењето на секој од роботите. Програмот ги користи модулите задршка на боја и хистерезис. Овие два модула се најидеални за најбрзо препознавање на позициите на секој од роботите. Програмот се извршува за секој од четирите роботи последователно. Овој програм е направен со цел да креираме виртуелна околина така да можеме и без сликата од камерата да ги управуваме роботите секој од нив поединечно. Крајниот корисник може и да не знае дека управува роботи туку да има претстава дека движи објекти во корисничкиот интерфејс. Предноста на ваквата визуелна претстава е да имаме можност да следиме огромен број на роботи во исто време. Овој програм може да се користи во видео надзорот

каде има голем број на камери и подвижните објекти ќе бидат претставени визуелно во посебен графички интерфејс. Во нашиот фудбалски куп на роботи наоѓа примена во прецизна селекција на саканиот робот играч. Доколку би биле поставени како некоја бројка тогаш ќе ни треба повеќе време за да го одбереме саканиот робот. Друга примена овој програм би нашол во следење на топката на некое фудбалско игралиште. Двата модула се сосема доволни за препознавање и следење на објектот како што е фудбалската топка. Исто така овој програм може да најде примена и во бранењето на голот од постигање на голови.

4.4 PHP скрипти

PHP е исклучително популарен јазик за скриптирање од општа намена и покрај фактот што беше формиран специјално за веб0развој. Во основа има синтакса многу слична со синтаксата на C, Java и Perl, но тоа е далеку поедноставно во однос на овие јазици со таа предност што е отворен код. Најчесто PHP скриптата е вградена во HTML кодот како внатрешни специјални тагови <?php ?>. Во моментот на активирање на документ кој има PHP скрипти на серверска страна кодот се извршува, а корисникот добива на пребарувачот чист HTML. Така PHP скриптата ги решава сите проблеми кои се карактеристични за типични CGI-апликации. Сепак PHP може да се користи и на локална клиентска страна и таа е пред сѐ главната примена на оваа технологија. Најпрво креирате скрипта која ќе се активира на серверска страна и ова е главната задача на PHP која најповеќе се користи во оваа верзија. Пишување на десктоп апликациите се извршуваат на клиентот (тука е потребно да се користи наставката PHP-GTK), но оваа верзија на PHP ретко се користи. PHP е крос платформска технологија. PHP дистрибуција е достапна за повеќето оперативни системи, вклучувајќи го Линукс, многу Unix (на пример, HP-UX, Соларис и OpenBSD), Microsoft Windows, Mac OS X, RISC оперативен систем и многу други. PHP го поддржуваат многу од најпопуларните веб сервер, како: Apache, Microsoft Internet Information Server, Microsoft Personal Web Server, Netscape, iPlanet, Oreilly Website Pro, Caudium, Xitami, OmniHTTPd

и други. За поголемиот дел од серверите PHP има 2 верзии - како модул и како CGI препроцесор. Покрај тоа на програмирање во PHP може да му се даде предност на двете процедурални и објектно-ориентирани програмирања (особено кога се работи со PHP 5). PHP е во состојба да генерира не само HTML документи, но, исто така, слики од различни формати, PDF и Flash. PHP е во состојба да генерира податоци во текст формат, вклучувајќи XHTML и XML. PHP поддржува ODBC и голем број на бази на податоци: Adabas D, InterBase, PostgreSQL, dBase, FrontBase, SQLite, Empress, mySQL, Solid, FilePro, Direct MS-SQL, Sybase, Hyperwave, MySQL, Velocis, IBM DB2, ODBC, Unix,Dbm, Informix, Oracle, DBX, Ingres, Ovrimos. Можете да креирате PHP скрипта да работи со протоколите LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформи за Windows), WDDX и многу други. PHP вклучува обработка на текст карактеристики како регуларни изрази perl, POSIX и напредни парсери за документи како XML, за кои PHP 4 користи стандарди саксофон и ДОМ, конвертира XML документи со наставката XSLT. Во PHP 5 XML обработката на документите се темели на користење на библиотеката libxml2 со кои им се додава две нови наставки: SimpleXML и XMLReader. За да го користите PHP во електронската трговија посебно во функција на правење исплати како што се Cybercash, CyberMUT, VeriSign Payflow Про и CCVS. Дополнително PHP поддржува многу други екстензии, како што се во функција на пребарувач mnoGoSearch, функциите на IRC Gateway, функции за работа со компресирани датотеки (gzip, bz2), календар со функција и компјутерски функции на превод, итн. И покрај фактот дека оваа технологија е едноставна можностите се огромни. Можете да го користите да се создаде произволно моќен, со високи перформанси веб-апликации кои ќе бидат во функција на сите барања на клиентите. Врската помеѓу програмот за следење на роботите, базата и програмот за приказ на роботите се остварува преку php скрипти преку кои правиме конекција до саканата база во која е креирана табела и во неа ги запишуваме сите податоци за потоа да ги читаме податоците т.е. вредностите на координатите на секој од роботите посебно и зададената команда на секој од роботите посебно. За запишување на податоците од моменталните позиции на роботите ја користиме скриптата „insert.php“ која пристапува до базата на податоци и во неа преку читање на пост методата ги запишува во базата mysql во соодветната табела од неа. За секој од роботите

правиме посебна скрипта преку која ги запишуваме соодветните информации за секој од роботите. Ова се прави пред сè за да се исклучи можноста ако еден робот се исклучи од врска со серверот тоа да не влијае на останатите роботи. Со ова се зголемува стабилноста на системот и може без разлика на тоа каква грешка настанала на еден од роботите тоа да не влијае на останатите роботи и воопшто на целиот систем.

Пример 3. PHP скрипти за комуникација со mysql база

insert.php

```
<?php
$player1x = $_POST['player1x'];
$player1y = $_POST['player1y'];
$player2x = $_POST['player2x'];
$player2y = $_POST['player2y'];
$player3x = $_POST['player3x'];
$player3y = $_POST['player3y'];
$player4x = $_POST['player4x'];
$player4y = $_POST['player4y'];
$con = mysql_connect("localhost","game","gamegame");
if (!$con)
 {
 die('Could not connect: ' . mysql_error());
 }
mysql_select_db("game", $con);
mysql_query("UPDATE position SET id='1', player1x = $player1x, player1y = $player1y, player2x = $player2x, player2y = $player2y, player3x = $player3x, player3y = $player3y, player4x = $player4x, player4y = $player4y");

mysql_close($con); ?
```

insert1.php

```
<?php
$player1free = $_POST['player1free'];
$player1command = $_POST['player1command'];
$con = mysql_connect("localhost","game","gamegame");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("game", $con);
mysql_query("UPDATE position SET id='1', player1free = $player1free,
player1command = $player1command");
mysql_close($con);
?>
```

Inser2.php

```
<?php
$player2free = $_POST['player2free'];
$player2command = $_POST['player2command'];
$con = mysql_connect("localhost","game","gamegame");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("game", $con);
mysql_query("UPDATE position SET id='1', player2free = $player2free,
player2command = $player2command");
mysql_close($con);
?>
```

insert3.php

```
<?php
$player3free = $_POST['player3free'];
$player3command = $_POST['player3command'];
$con = mysql_connect("localhost","game","gamegame");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("game", $con);
mysql_query("UPDATE position SET id='1', player3free = $player3free,
player3command = $player3command");
mysql_close($con);
?>
```

insert4.php

```
<?php
$player4free = $_POST['player4free'];
$player4command = $_POST['player4command'];
$con = mysql_connect("localhost","game","gamegame");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("game", $con);
mysql_query("UPDATE position SET id='1', player4free = $player4free,
player4command = $player4command");
mysql_close($con);
?>
```

read.php

```
<?php
mysql_pconnect ("localhost", "dimitrija_game", "gamegame");
mysql_select_db ("dimitrija_game");
$qResult = mysql_query ("SELECT * FROM position");
$row = mysql_fetch_array($qResult);
$rString
="&player1x=".$row['player1x']."&"."player1y=".$row['player1y']."&"."player2y=".$row['
player2y']."&"."player2x=".$row['player2x']."&"."player3y=".$row['player3y']."&"."player
3x=".$row['player3x']."&"."player4x=".$row['player4x']."&"."player4y=".$row['player4y']
."&"."player1command=".$row['player1command']."&"."player1free=".$row['player1fre
e']."&"."player2command=".$row['player2command']."&"."player2free=".$row['player2
free']."&"."player3command=".$row['player3command']."&"."player3free=".$row['play
er3free']."&"."player4command=".$row['player4command']."&"."player4free=".$row['pl
ayer4free']."&";
echo $rString;
?>
```

reset.php – скрипта за ресет

```
<?php
$con = mysql_connect("localhost","dimitrija_game","gamegame");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("dimitrija_game", $con);

mysql_query("UPDATE position SET id='1', player1free = '0', player1command = '0',
player2free = '0', player2command = '0', player3free = '0', player3command = '0',
player4free = '0', player4command = '0'");
mysql_close($con);
```

4.5 MySql база

Потребата од користење на база е од особена важност бидејќи сите повици кои ги правиме до неа се опслужуваат временски. Ако користиме некој текст фајл и во него ги запишуваме податоците ќе настане доколку има два повици во исто време еден од нив нема да се изврши. Затоа најдоброто решение е користење на база кај која драјверот ќе ги извршува сите команди. За да може да имаме успешно запишување во базата треба во неа претходно да имаме креирано табела „position“ во која ќе ги чуваме координатите од роботите и командите кои ги извршуваат моментално. Сите вредности се од тип цел број со почетна поставена вредност на нула. Креираме табела со следната скрипта во која ќе ги чуваме сите податоци за секој од роботите.

Скрипта во mysql јазик со наслов game.sql

```
CREATE TABLE IF NOT EXISTS position (
  id bigint(10) NOT NULL AUTO_INCREMENT,
player1 int(11) NOT NULL DEFAULT '0',
  player1y int(11) NOT NULL DEFAULT '0',
  player2x int(11) NOT NULL DEFAULT '0',
  player2y int(11) NOT NULL DEFAULT '0',
  player3x int(11) NOT NULL DEFAULT '0',
  player3y int(11) NOT NULL DEFAULT '0',
  player4x int(11) NOT NULL DEFAULT '0',
  player4y int(11) NOT NULL DEFAULT '0',
  player1command int(11) NOT NULL DEFAULT '0',
  player1free int(11) NOT NULL DEFAULT '0',
  player2command int(11) NOT NULL DEFAULT '0',
  player2free int(11) NOT NULL DEFAULT '0',
  player3command int(11) NOT NULL DEFAULT '0',
  player3free int(11) NOT NULL DEFAULT '0',
  player4command int(11) NOT NULL DEFAULT '0',
  player4free int(11) NOT NULL DEFAULT '0',
```

```
  PRIMARY KEY (id)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=2 ;

INSERT INTO position (id, player1x, player1y, player2x, player2, player3x, player3y,
player4x, player4y, player1command, player1free, player2command, player2free,
player3command, player3free, player4command, player4free) VALUES
(1, 216, 149, 256, 70, 397, 34, 338, 115, 0, 0, 0, 0, 0, 0, 0, 0);
```

4.6 Програм за селекција на слободен робот

Во овој програм имаме визуелен приказ на секој од роботите и нивните моментални позиции. Соодветно секој од роботите е означен со посебна боја. Корисникот го одбира слободниот робот и со притискање на копчињата за лево, десно, горе, долу, копчето н и копчето м го управува саканиот робот. Ако е роботот претходно зафатен од друг корисник тогаш нема да може да го селектираме. Секој од објектите во програмот се иницијализира во секој нов фрејм на флеш програмот. Така да доколку некој од корисниците одлучи да го откаже движењето на работите во наредниот фрејм роботот ќе може да се селектира и понатаму да се управува. Десно од игралиштето е приказ на моменталниот статус на роботите. Во него се прикажани тековните команди кои се извршуваат и статус кој од роботите е слободен. На слика со ред број 33 е прикажан програмот за селекција и управување на роботите. Програмскиот код е оптимизиран во минимален број на редови за да се намали времето на праќање на командите од програмот на секој од роботите. Програмот постојано прави обновување на координатите на секој од роботите и нивниот статус. Податоците ги чита од базата која е поставена на серверска страна. Во неа се поставуваат вредностите за секој од роботите. Тие податоци доаѓаат од програмот за детекција и следење на роботите. Тој програм е поставен на локалниот компјутер кој е задолжен за извршување на командите кои стигаат од овој програм и паралелно полнење на базата со вредностите кои ги прави програмот за препознавање и детекција.

Верзија во која е компајлиран овој програм е Adobe Flash 8. Во овој програм може да имплементираме решенија кои ќе ги елиминираат колизиите кои ќе настануваат при интеракцијата помеѓу секој од роботите. На овој начин може да направиме кооперација помеѓу роботите без притоа корисниците да имаат удел во тоа сè со цел да се намалат ударите кои можат да настанат при нивно управување.

Слика 33. Програм за селекција и управување на саканиот робот

Figure 33. Program for the selection and management of the desired robot

4.7 Програм за контрола на робот преку Bluetooth конекција

Програмот legoonline.exe е за задавање команди на роботот преку читање на податоци од mysql база која е поставена на серверска страна. Во оваа база се чуваат податоци од координатите на секој од роботите и соодветно која команда е активирана. Секој од роботите за да работи независно од останатите направени се четири програми посебно legoonline1.exe, legoonline2.exe, legoonline3.exe и legoonline4.exe. Секој од овие програми се конфигурира поединечно преку посебен текст фајл “config.cfg”. Во него е поставена за секој од роботите посебно порта. Оваа порта ја добиваме кога правиме спарување на секој од роботите со локалниот компјутер. Комуникациската библиотека во c++ дозволува на лесен начин да комуницираме со роботот Lego Nxt преку Bluetooth комуникација или мрежна конекција. За да може да комуницираме со Lego Nxt роботот преку Bluetooth комуникација потребно е претходно да го инсталираме драјверот за уредот и да ги пријавиме сите роботи на кои им се вклучени уредите за безжична комуникација во нашиот случај Bluetooth уредот. Еднаш кога ќе се воспостави врска може да ги видиме кои порти ги користи саканиот робот и со тоа во конфигурацискиот фајл да ги вметнеме истите. На овој начин при стартување на програмите посебно секој робот ќе се иницијализира на посебна порта и нема да има мешање на командите. Роботите претходно се пријавуваат со лозинка за да се увериме дека се работи за саканото спарување. Оваа C++ комуникациската библиотека може да се користи во најголем дел од c++ компајлерите на Windows оперативните системи. Исто така овозможено е и на џебните компјутери кои користат Windows Mobile. При тоа за програмирање и компајлирање го користиме програмот MS Visual Embedded C++.

Можностите на оваа библиотека се следните:

- Отворање и затворање на конекција со повеќе роботи истовремено.
- Контрола на моторите .
- Праќање и примање на пораки користејќи го маилбоксот.

- Користење на стандардни LEGO вклучувајќи ги звучните и сонарните сензори.
- Поставување на име на лего коцката, земање на нивото на батеријата, читање на која верзија користи лего коцката и друго.
- Свирење на некој тон.
- Свирење на некоја музика.
- Користење на NXT's фајловиот систем.
- Симнување и праќање на фајлови.
- Стартување и стопирање на програми во лего коцката.
- Користење на компас, жироскоп, сензор за боја и тилт сензор од фирмата Hitechnic.
- Комуникација со I^2C сензорите.
- Користење на I^2C PCF8591 A/D конверторот.
- Користење на I^2C PCF8574 I/O чипот.
- Користење на исклучоци за фаќање на некој настан од некој сензор и конекциски грешки.
- Генерички конекциска класа преку која ќе може да ја смените конекцијата помеѓу Bluetooth или мрежна комуникација, и уште многу други можности.

Табела 2. Опис и намена на библиотеките
Table 2. Description and purpose of libraries

NXT C++ Bluetooth библиотеки NXT C++ Bluetooth library	Опис и намена на библиотеката Description and purpose of the library
Class Summary	
Adc_8591	Class for the PCF8591 8-bit A/D D/A converter (D/A feurures not implemented yet!)
Adc_ports	Class to retrieve ADC port reading from the the PCF8591 ADC chip/sensor

Bluetooth	Class for Bluetooth communication
Brick	Class to get/set brick name, get/set battery level, R/W to mailbox, start/stop programs, play sound, stop sound ect.
Color_sensor	Class for HiTechnic color sensor
Compass	Class for HiTechnic compass sensor
Connection	Abstract class for connections
Connection_type	Connection type enumeration
Device_info	typedef for Device_info_t
Device_info_t	Struct used to retriece device info
Device_version	typedef for Device_version_t
Device_version_t	Struct uesd to retriece device version
Error_type	Error type enumeration
File_mode	Enumeration for file modes when opening a file
Filesystem	Class to interact with the filesystem on the NXT
Gyro	Class for HiTechnic Gyro sensor
I2c	Abstract class for digital I2C sensors
Io_8574	Class for the PCF8574 8-bit I/O I2C chip
Led_mode	Enumeration for different NXT light sensor LED modes
Light	Class for NXT light sensor
Light_rcx	Class for RCX light sensor
Motor	Class for the NXT motors
Motor_port	Motor port enumeration
Nxt_errors	Error code enumeration (error code list)
Nxt_exception	Class used to throw exceptions when communication fails or the NXT reports an error
Nxt_file	typedef for Nxt_file_t
Nxt_file_t	Struct that holds information on a nxt file
Nxt_network	Class for Network communication
Result	Class used to return sensor readings when there is more than one return value
Result_type	Enumeration used to determine the result type

	(which sub-class of Result)
Rgb_color	Class to retrieve red green and blue colors from the HiTechnic color sensor
Rotation	Class for RCX rotation sensor
Sensor	Class for analog sensors - also works as the base class for all other sensors
Sensor_mode	Sensor mode enumeration
Sensor_port	Sensor port enumeration
Sensor_type	Sensor type enumeration
Server_mode	Network server mode enumeration
Server_settings	typedef for Server_settings_t
Server_settings_t	Struct used to retriece server settings when a network connection is used
Sonar	Class for the NXT sonar (ultrasonic) sensor
Sonar_mode	Enumeration for different sonar sensor modes
Sound	Class for NXT sound sensor
Sound_mode	Enumeration for different NXT sound sensor modes
Temp_mode	Enumeration for different temperarure sensor modes
Temperature	Class for RCX temperature sensor
Tilt	Class for HiTechnic tilt/accekeration sensor
Touch	Class for NXT and RCX touch sensor
Xyz_position	Class to retrieve XYZ coordinates from the HiTechnic Tilt/Accekeration sensor

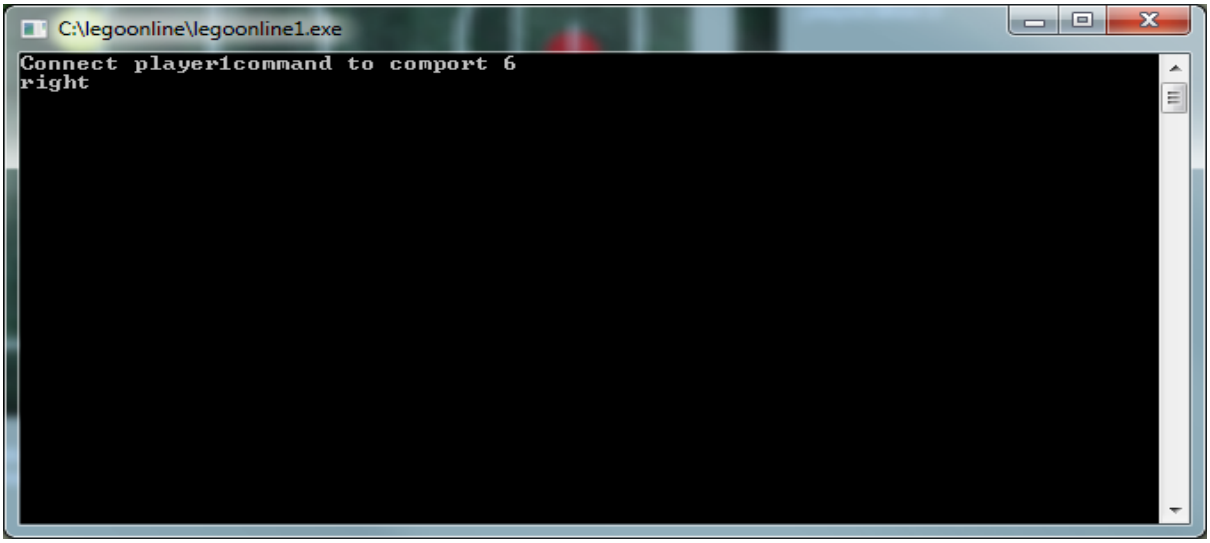
Во конфигурациски фајл **config.cfg** ќе се чуваат информациите за тоа кој робот кој порт ќе го користи за да се исклучи можноста од мешање на командите.

Пример 4. Фајл за иницијализација на портот на секој од роботите

config.cfg

player1command 6 player2command 16 player3command 20 player4command 28

Програмот се компајлира во dev-cpp како конзола апликација направен е пред сè за користење на комуникациските библиотеки на c++ т.е. праќање на команди на фиксен порт кој го користи зададениот робот. Во главниот програм ги вклучуваме сите библиотеки кои ќе ги користиме и после тоа правиме конекција со име Bluetooth исто така правиме за секој мотор посебно конекција.



Слика 34. Приказ на примената команда од серверска страна преку пост методата

Figure 34. A review of the application command from the server side through the post method

Слика 35. Приказ на примената команда од серверска страна преку постметодата

Figure 35. A review of the application command from the server side through the post method

5. РЕЗУЛТАТИ

За да може правилно да оцениме колку алгоритмите на препознавање и на задавањето на командите брзо се извршуваат во програмот legoonline.exe имплементиран е код кој ја користи постметодата за праќање на команди преку php скрипта. Ова е направено сè со цел да се елиминира влијанието на човечкиот фактор при тестирањето со тоа и ненамерното доцнење при задавање на командите од клиентот кој го врши тестирањето. Резултатите се прикажани во табелата со реден број 3. Во неа ги запишуваме вредностите добиени по експериментален пат со реалните роботи.

5.1. Резултати од испитувања на оддалечени сервери

За да извршиме правилно тестирање последователно праќаме две команди за лево и во наредниот момент за десно и броиме колку пати се извршуваат командите во една минута при самостојна работа на програмот. Во програмот ја користиме библиотеката „libcurl.dll“ преку која се извршува постметодата т.е. ги пренесуваме локалните варијабли одвоени со амперсанд помеѓу себе прикажано во извадок од кодот прикажан подолу. Оваа библиотека е најчесто користена од програмерите бидејќи се одликува со голема стабилност, користи минимални хардверски ресурси и е користена во голем број на апликации. Во овој пример ќе покажеме како се користи оваа библиотека „libcurl.dll“ за правилно праќање на команда која треба да се изврши на роботот со реден број еден и задавање на команда да се врти налево круг.

Пример 1. Користење на библиотеката libcurl.dll во c++ програм

```
CURL *curl;
CURLcode res;
int temppp = 0;

struct curl_httppost *formpost=NULL;
```

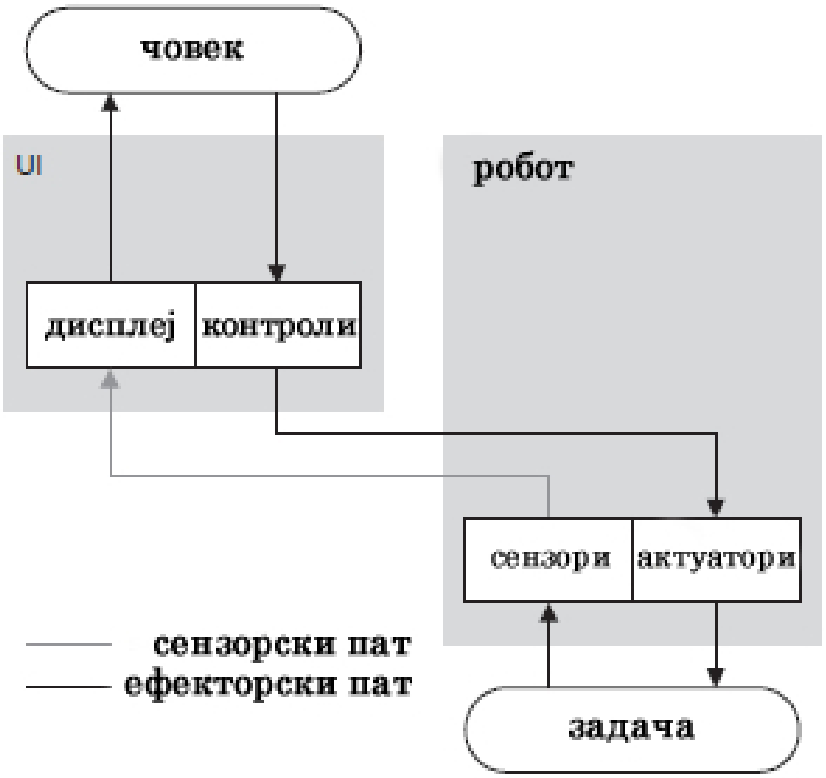
```
  struct curl_httppost *lastptr=NULL;
  struct curl_slist *headerlist=NULL;
  static const char buf[] = "Expect:";
 curl_global_init(CURL_GLOBAL_ALL);
 curl_formadd(&formpost,&lastptr,CURLFORM_COPYNAME,
"submit",CURLFORM_COPYCONTENTS, "send",CURLFORM_END);
 curl = curl_easy_init();
 headerlist = curl_slist_append(headerlist, buf);
 curl_easy_setopt(curl,                                    CURLOPT_URL,
"http://robocuponline.org/insert1.php");

curl_easy_setopt(curl,CURLOPT_POSTFIELDS,"player1free=0&player1command=4
");
    res = curl_easy_perform(curl);
    curl_easy_cleanup(curl);
    curl_formfree(formpost);
    curl_slist_free_all (headerlist);
```

Со ова постигнуваме праќање на командите од самиот програм до базата и истиот програм симултано ги чита командите и потоа ги извршува.

5.2 Реакција на роботите на различни терени

Секој од роботите има тенденција на зададената наредба за најкратко време да ја постигне зададената вредност. Доколку дојде на пречка вградениот енкодер ќе врати вредност која покажува дека сме дошле на пречка и може да врати команда дека не може да ја изврши командата. Затоа од особена важност се вградените енкодери кои заедно со електричниот мотор прават еден сервомотор кој има повратна врска. Со ова се елиминира можноста од неизвршување на командата бидејќи повратната врска ја елиминира секоја можност од неизвршување на командата. Во зависност од теренот користиме различни типови на тркала и придвижувачки нозе. На сликите со реден број 8 и 9 прикажани се возила со необични тркала кои може да се движат на нерамни терени. На слика со реден број 10 прикажан е робот кој може да се движи на

тешки непроодени терени. За да може роботите да се движат беспрекорно добро на самиот почеток најчеста метода која се користи за управување на далечина е директната метода. Таа е прикажана на сликата со реден број 29 каде имаме директна врска помеѓу роботот и човекот. За да се избегне неизвршување на командата користиме комбинирана врска каде сензорите кои се вградени во роботот на самиот микропроцесорски систем ќе дадат повратна информација дека командата не е извршен и микропроцесорскиот систем потоа ќе го започне процесот на извршувањето на командата. Така со овој комбиниран метод добиваме намалување на времето за извршување на командата. Корисникот кој управува со роботот еднаш ја задавува командата. Со тоа се елиминира трошење на дополнително време за извршување на зададената задача до роботот.

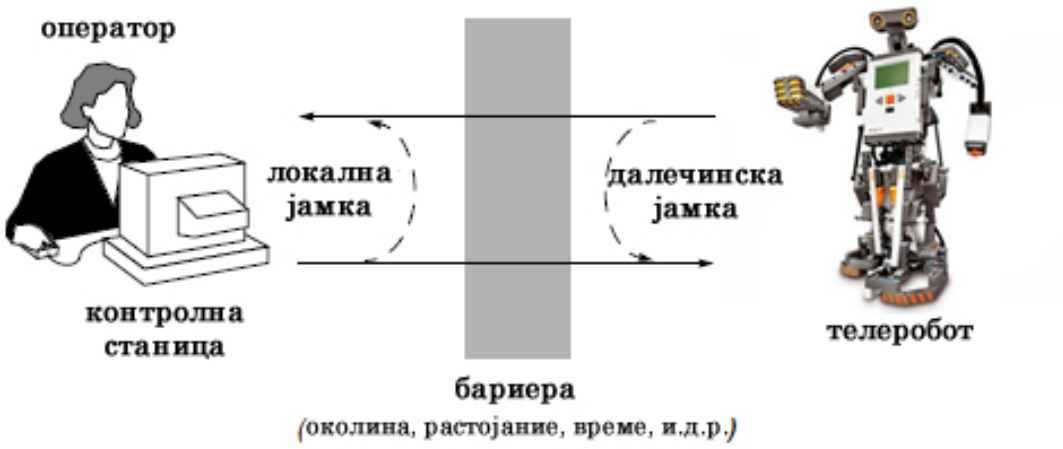


Слика 36. Директна контрола
Figure 36. Direct Control

Ја користиме оваа метода бидејќи е лесна за имплементација, во нашиот систем ги користиме стрелките од тастатурата за давање на насока на движење, а визуелно имаме приказ на позицијата на секој од роботите. Со директната контрола човекот ја затвора јамката. Ова често претставува проблем бидејќи сѐ зависи од човекот кој го управува роботот. Доколку се случи некој од роботите да го истурка другиот робот тоа ќе продуцира со промена на координатите на другиот робот. Ова може да се избегне со тоа што ќе вградиме ултразвучни сензори на секој од роботите. Доколку дојде до некое минимално растојание од останатите роботи се прекинува движењето во таа насока. На тој начин ќе се избегне можност од судри и прекини на акцијата. Друго решение е доколку дојде до колизија во главниот програм за следење на роботите по боја да се даде автоматски команда за ограничување на движењето на роботите во таа насока. Со ова ќе остваримe комбинирана помошна контрола помеѓу роботите.

5.3. Резултати од лабораториски испитувања

Исполнување на барањата, потреби и очекувања поставени од страна на корисниците е прв приоритет во овој магистерски труд. За системот да биде широко употреблив мора да помине успешно низ повеќе тестови. За таа цел правиме испитувања на системот во сите можни околини и донесуваме заклучоци врз база на добиените вредности од мерењата. Локалните команди се извршуваат со огромна брзина и може да опслужат онолку роботи колку што имаме направено. Брзината на извршување на командите може да ја измериме по експериментален пат. На сликата под реден број 37, е прикажана бариерата која постои помеѓу локална и далечинска јамка.



Слика 37. Бариера помеѓу извршувањето на командите
Figure 37. Barrier between execution of commands

Користиме метод на опсервација и гледаме колку пати роботот си ја менува насоката на ротација во единица време. За да добиеме точни резултати правиме повеќе тестови и резултатите ги запишуваме во табела. Од експерименталните резултати констатирано е дека за време од една минута се извршуваат просечно по 96 команди или средна вредност за една команда да се изврши е 0.623 секунди во просек. Ова е добиено при правење на вкупно 10 теста. Сите тестови се правени на секој од четирите роботи поединечно и добиени се следните резултати кои се евидентирани во табела со реден број 3.

Табела реден број 3. Вредности од времето на извршување на командите
Table ordinal number 3. Recorded values of time execution commands

Реден број на тест/ Ordinal test	Вкупен број на успешно извршени команди во една минута / Total number of successfully executed commands in a minute	Просечно време на извршување на една команда во една секунда(s) / Average time of execution of a command in one second(s)	Разлика од средната вредност на извршување / Unlike the mean execution	Средна вредност на извршување / Mean execution

1	95	0.631	0.008	0.623
2	98	0.612	-0.011	0.623
3	95	0.631	0.008	0.623
4	96	0.625	0.002	0.623
5	97	0.618	-0.005	0.623
6	96	0.625	0.002	0.623
7	95	0.631	0.008	0.623
8	98	0.612	-0.011	0.623
9	97	0.618	-0.005	0.623
10	96	0.625	0.002	0.623

Од табелата со реден број 3 може да заклучиме дека постои мала временска разлика која е максимум плус - минус 11 милисекунди. Ова е сосем задоволително отстапување за фудбалскиот куп на роботи. Целиот систем е динамичен и постојано се менуваат сите позиции на роботите. За да го намалиме времето на извршување на командите потребно е серверот да биде што поблиску до местото каде се извршуваат сите настани, или да користиме што побрз интернет.

6. ДИСКУСИЈА

На роботиката како што ја имаме во сегашно време и претстои дол пат полн со нови предизвици исполнет со конкретни решенија и широка примена во секојдневниот живот. На почеток роботите беа со огромни димензии поврзани со големи кабли и скапи интерфејси. Во денешно време роботите се прават во мали размери со можност за управување преку мали портабли уреди. Целата контрола на роботите ја остваруваме со мобилните уреди бидејќи тие имаат интернет врска преку мобилните оператори. Во овој магистерски труд давам општ преглед на теле роботиката во поглед на историски и сегашен ден. Креирани се програми кои се извршуваат на најголем дел од денешните компјутери било клиентски или серверски. Овие програми користат мрежа која е широко користена. Ова поле во роботите е сè уште во развој и секојдневно се

подобруваат решенијата кои имаат голема потреба за ваквите системи. Овој систем со сите решенија кои се имплементирани во него може да најде примена во медицината, истражување на вселената, подводните истражувања и во многу други домени на технологијата. Идеите во овој труд се експериментално проверени и веќе најдоа примена во фудбалскиот куп на роботи. Теле роботиката наоѓа постојано секојдневна примена и таа не може да има прогрес сама туку мора да работи на мултидисциплинарен начин. Може слободно да се увериме дека оваа технологија сама не може да ги оствари соништа на роботите. На оваа технологија и е потребна интеграција со останатите технологии. На теле роботиката и се потребни сите знаења, техники и методи кои ги располага човештвото. Историчарите на роботиката се согласуваат дека за прв пат јавно е објавен зборот робот во 1921 година. Овој збор беше објавен од чешкиот писател Капек кој во неговиот текст R.U.R (Rossum's Universal Robots) ги објаснува вештачките луѓе. Фактичката референца доаѓа од повеќе официјални и неофицијални историчари на роботите или слични уреди на нив. Основното прашање кое треба да си го поставиме е: Што треба да направиме за да остваримe одлична соработка помеѓу човекот и роботот кој се управува? Одговорот на ова прашање е комплициран со оглед на фактот дека за успешното функционирање на системот зависат многу фактори. Најважниот фактор е брзината на пренос на информациите преку интернетот. Комуникациските протоколи сосема ги задоволуваат софтверските решенија кои се имплементирани во овој систем. Главниот дел во кој треба да се фокусираме е самиот робот и бројот на сензори кои треба да ги поседува за правилно да ги извршува сите команди кои се последица од директната контрола. За да го намалиме времето на доцнење потребно е да користиме комбинирани протоколи. Во зависност од специфичните потреби треба да се активираат побрзите протоколи сѐ со цел времето на доцнење да се сведи на минимум. „Lego Mindstorms NXT“ е програмирачки робот кој е направен од Lego во јули 2006 година, заменувајќи ја првата верзија „Lego Mindostorms“ . Кутијата содржи 519 коцки, 3 пренесувачки мотори, 4 сензори (ултрасоничен, звучен, допир и светлина), 7 кабли за поврзување, USB кабел и NXT програмирачки контролор. Контролорот е познат и како „мозокот“ на Mindtorms машините. Тој му дозволува на роботот да прави разни операции. Кутијата исто така содржи и NXT-G, графичка програма за

програмирање која овозможува креирање и преземање на програми на NXT (главниот контролор). Lego Mindstorms NXT 2.0„Lego Mindstorms NXT 2.0“ е објавена на 5 август 2009 година. Кутијата содржи 619 коцки (вклучувајќи ги и сензорите и моторите), новиот сензор за боја, два сензора за допир и ултрасоничен сензор. Овој робот потполно ги задоволува потребите за успешна реализација на директната контрола. Се одликува со голема стабилност и може да се надградува во кој било правец. Со вградениот комуникациски модул добиваме повратна врска од роботот. На тој начин ја добиваме моменталната состојба на секој составен елемент од роботот. Безжичната комуникација е најдобриот начин за праќање и примање на информациите со роботот. Тоа е последица од сигурните протоколи кои се имплементирани во комуникацискиот модул. Најголемиот интерес за мобилните роботи го направија научно фантастичните филмови во кои роботите се претставени како напредни интелигентни машини по особини и карактер слични на човекот. Мобилните роботи претставуваат инспирација и задоволство за студентите ширум сите универзитети во светот. Мобилните роботи бидејќи имаат подвижен карактер и плус со можност за далечинска контрола се вистински предизвик за идните истражувачи, конструктори и програмери.

7. ЗАКЛУЧОК

Ние секојдневно правиме интеракција со машините и таканаречените човечки креирани ентитети. Ја ставаме картичката, притискаме на копчињата и ги следиме инструкциите од апаратот за подигнување пари. Ја вклучуваме машината за перење или микробрановата печка скоро секој ден. Мора да си признаеме дека ова е ера на машини изработени од човекот кои се интерактивни и лесни за ракување. Вистинското прашање е како човекот ги перципира роботите и каков е сегашниот однос и релацијата помеѓу човекот и машината. Денес роботиката зазема сè позначајно место во животот и работата на човекот. На почетокот на 21 век започна експанзија на практична примена на роботиката и интелигентните машини. За да биде човекот доволно информиран во техничкото опкружување и да може подобро да го користи потребно е да биде доволно технички образован во оваа област. Ова образование мора да биде флексибилно со постојана надградба. Затоа е од особена важност улогата на високообразовните институции кои мораат да создадат образовни содржини од роботиката кои ќе обезбедат минимум потребно знаење од оваа област. Овој магистерски труд ги содржи основните технологии кои треба да се користат за да може да имаме управување на роботите и интелигентните машини на далечина. Еден од поголемите предизвици во развојот на вештачката интелигенција е соработката помеѓу робот и човек. Соработката за разлика од интеракцијата подразбира процес на интелигентно предвидување на следните активности на соработникот, и дејствување во согласност со тие очекувања. Притоа можно е да се поддржи соработникот или да му се укаже на поинакво мислење, со цел да се избегнат грешки и да се дојде до најдобро консензуално мислење во екипата. Затоа соработката подразбира добро познавање на соработниците, градење на екипата, сочувствување, чувство за заеднички успех и сл., што е многу повеќе од проста интеракција преку човечките или роботските сетила. Оттука интелигенцијата која треба да се манифестира за да се обезбеди успешна соработка при решавање на сложени проблеми често пати е тешко достижна и за луѓето, и затоа претставува голем предизвик во развојот на вештачката интелигенција. Со оглед на фактот дека роботската индустрија се развива

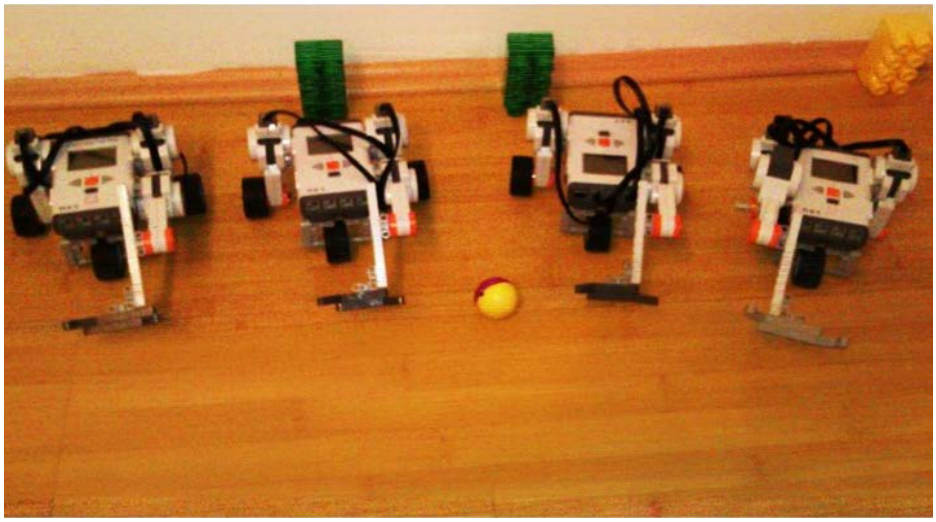
исклучително брзо, мошне значајно е да се дефинираат правила за соработка помеѓу роботските системи, со цел да се пресретнат голем број етички прашања за заедничкото живеење на луѓето и роботите во хармонија. Истражувачите најчесто ги гледаат мобилните роботи како квази статички уреди. Бројни примери на роботи со четири, шест или повеќе тркала се развивани така да ја максимизираат мобилноста на лоши терени. За разлика од овие егзоскелетните роботи кои можеби имаат и поголем потенцијал за мобилност, се изградени и демонстрирани главно така да имаат ниско поставено тежиште и плоча како потпорна база, која заедно со интелигентен контролен алгоритам е дизајнирана да го задржува гравитациониот вектор на тежиштето низ базичниот полигон (т.е. нагибна моторика, ножна координација). Кај голем број концепти направен е обид за максимизирање на мобилноста со помош на големи тркала или нозе, со проширени обрачи на тркалата или стапалата на нозете, повеќе тркален двигател, голем тело/површина слободен простор, артикулирани конфигурации на телото итн. Ваквите роботи најчесто се ограничени во смисла на планирана мобилност, така што се дизајнирани за оперирање на мали брзини, кои типично достигнуваат до 1 км/ч или помалку. Динамичките фактори имаат мало влијание на ваквите системи, и поради тоа најчесто се игнорираат. За надминување на ваквите ограничувања во роботиката мора да се користат контролори со кои ќе се надминуваат постепено проблемите. Еден од многуте контролори кои се користат моментално е EyeBot контролорот за мобилни роботи со тркала, роботи што одат или летачки роботи. Со овој контролор ни се овозможува да им ја зголемиме мобилноста, интелигентноста на мобилните роботи. Преку програмот за симулација на развиените контролори многу брзо ги откриваме грешките при развивањето на соодветен алгоритам со тоа сме ослободени од непотребно фрлање на средства за роботи за кои не сме сигурни дали постојниот програм ќе е соодветен за физичкиот робот. Со сегашното ниво за поддршка EyeBot и Lego Nxt е во водечките мобилни роботски платформи најчесто поради неговата директна врска со камерата и вградените безжични модул за комуникација со останатите уреди. Тоа се должи на тоа бидејќи камерата игра многу важна улога во решавање на проблемите од роботиката посебно во фудбалскиот куп на роботи. Преку практичната симулација на робокупот се гледа неговиот огромен потенцијал во успешно решавање на

проблемите во голем дел на динамичките автоматски системи. Друго што треба да споменеме е дека технологиите што се користени во овој магистерски труд се од отворен карактер и може да се користат на повеќе платформски систем. Во овој магистерски труд на веб-страната “www.robotsonline.info” е поставен и програмскиот код кој е од отворен карактер. Овој програм е слободен за употреба и развој исклучиво во научно истражувачки цели.

8. ДОДАТОК

Во историјата на вештачката интелигенција и роботиката годината 1997 ќе се запамети како пресвртна точка, во мај 1997 година компјутерот IBM Deep Blue го победи светскиот шампион во шах. Четиринаесет години во постојана битка со човекот компјутерот конечно извојува победа. На 4 јули 1997 година роботот на Наса „pathfinder mission“ направи успешно слетување на црвената планета Марс и беше првиот автономен роботски систем кој успешно се движи на планетата. Овој роботски куп онлајн претставува почеток на креирање и развој на роботски системи кои ќе го победат Светскиот шампион во фудбал. Целта на овој магистерски труд е развој на систем кој ќе биде широко применет. За таа цел во овој додаток се вметнати слики кои покажуваат четири роботи придвижувани со два сервомотори. Целта им е да го движат топчето и да го удираат со третиот сервомотор кој е поставен на левата страна од роботот. На сликите под реден број 38, 39 и 40 има поставено илузорно коцки кои ги претставуваат краевите на игралиштето и секој од головите соодветно. При конструкција на овој робот користени се минимален број на елементи потребни за извршување на сите операции при играњето на фудбал. На слика со реден број 41 е прикажан најпознатиот микропроцесор M68332 на фирмата „моторола“ која развива стабилни микропроцесори кои се одликуваат со огромна процесирачка моќ. Овој процесор е 32 битен и во себе е задржана компатибилноста на програмите направени за 8 битна архитектура за да можат да се извршуваат на овој микропроцесор. На следниот линк може да се преземе комплетната документација и комплетниот софтвер за роботот EyeBot http://robotics.ee.uwa.edu.au/eyebot/. Голем дел од корисниците на интернетот

имаат потреба да управуваат роботи и настани преку интернет на интерактивен начин. Најпрво тоа се користи за забава за потоа со тек на време да се искористи и прерасне во управување и надзор на својот дом, работно место, работилници и фабрики поточно секаде каде што има потреба од активирање и следење на настани. На сликите под реден број 38, 39, 40 се прикажани роботите во реално време како движат топче и го шутираат на импровизиран гол. Најпрво тоа се користи како забава, а потоа со тек на време се зголемува и интересот на кој начин ваквите системи функционираат и како сето тоа може да се искористат за свои лични потреби и потребите на останатите.

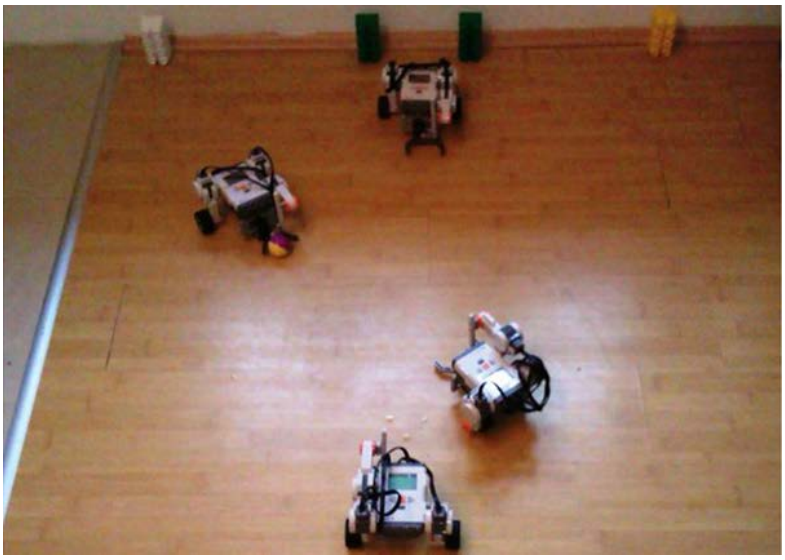


Слика 38. Приказ на роботи во акција

Figure 38. Display of robots in action

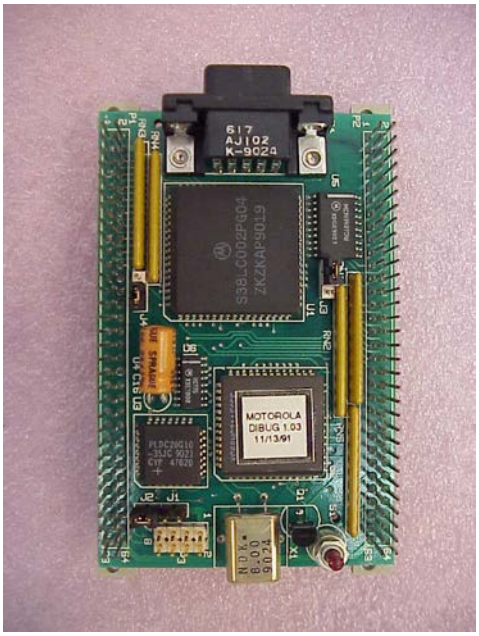
Слика 39. Приказ на роботи во акција

Figure 39. Display of robots in action



Слика 40. Приказ на роботи во акција

Figure 40. Display of robots in action



Слика 41. Развојна плоча за микроконтролерот M68332BCCDI Rev C

Figure 41. Development Board Microcontroller M68332BCCDI Rev

Слика 42 Модул LAN CONTROLLER

Figure 42. Module LAN CONTROLLER

Овој модул кој е прикажан на слика по реден број 42, претставува мал веб-сервер преку кој се пренесуваат информациите од влезовите и се задаваат команди на излезите од самиот модул. Овој модул содржи еден микроконтролор преку кој се генерираат сите html страни. Содржи 5 аналогни влеза, 4 дигитални влеза и 5 излези. Преку овој модул на брз и едноставен начин се врши управување и следење на настаните во системите и работните околини. На слика со реден број 43 прикажани се повеќе роботи кои играат фудбал помеѓу себе. Вградената камера и вградениот безжичен комуникациски модул се главните елементи на овие роботи. Сликата која се добива од роботите се праќа на серверска страна, а од таму се враќа соодветно дејство на роботите. Така да со ова продуцираме програм кој се извршува на далечина и кој постојано трпи софтверски подобрувања во поглед на имплементација на подобри решенија за детекција на објектите и реализирање на соодветна стратегија во активното постигнување на голови.

Слика 43. Приказ на Eyebot роботи како играат фудбал

Figure 43. Figure Eyebot robots playing football

9. КОРИСТЕНА ЛИТЕРАТУРА (REFERENCES)

[1] Giulio Ferrari, Andy Gombos, Soren Hilmer, Jurgen Stuber, Mick Porter, Jamie Waldinger, LEGO MINDSTORMS NXT G Programming Guide 2nd Edition Jun 2010

[2] M. Micire, J.L. Drury, B.Keyes, H. A. Yanco, Multi-touch interaction for robot control, IUI '09 Proceedings of the 14th international conference on Intelligent user interfaces, ACM New York, NY, USA 2009

[3] S. Yang, R.R. Igorevich, Y. Park, D. Min, E. Choi, Reliable robot teleoperation service architecture for various controllers and robots, In the Proc. of the 7th International Conference oNetworked Computing and Advanced Information Management (NCM), 2011

[4] S. Koceski, N. Koceska, Vision-based gesture recognition for human-computer interaction and mobile robot's freight ramp control, 32nd International Conference on Information Technology Interfaces (ITI), 2010

[5] Baohua Tan, A Distributed Speech Romote Control System Based on Web Service and Automatic Speech Recognition, Electrical Power Systems and Computers LNEE, Volume 99, 771-778, 2011

[6] D. Balakrishna, P. Sailaja, R.V.V.P.Rao, B. Indurkhya, A novel human robot interaction using the Wiimote, In the Proc Of IEEE International Conference on Robotics and Biomimetics (ROBIO), 2010

[7] C. Smith, H.I. Christensen, Wiimote robot control using human motion models, In the Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009.

[8] T. Schlimer, B. Poppinga, N. Henze, S. Boll, Gesture recognition with a wii controller. In TEI ’08: Proceedings of the 2nd international conference on Tangible and embedded interaction, ACM, New York, NY, USA, TEI, 11–14, 2008

[9] Using Mobile-phone Accelerometer for Gestural Control of Soccer Robots Mediterranean Conference on Embedded Computing MECO - 2012 Bar, Montenegro Kire Serafimov, Dimitrija Angelkov, Natasa Koceska, Saso Koceski

[10] C. Smith, H.I. Christensen, Wiimote robot control using human motion models, In the Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009.

[10] ITU Corporate Annual Report. (2009). http://www.itu.int/dms_pub/itu-s/opb/conf/S-CONF-AREP-2008-E06-PDF-E.pdf (Online, visited 10.05.2012).
[11] Sakamoto, D., Honda, K., Inami, M., Igarashi, T.: Sketch and Run: A Stroke-based Interface for Home Robots. In: 27th International Conference on Human Factors in Computing Systems, pp. 197--200, (2009)
[12] Smartphone forecast by Pyramid Research, http://www.pyr.com/pr_prlist/PR121009_SMART.htm (2009)
[13] Kemp, C.C., Anderson, C.D., Nguyen, H., Trevor, A.J., Xu, Z.: A Point-and-Click Interface for the Real World: Laser Designation of Objects for Mobile Manipulation. In: 3rd ACM/IEEE International Conference on Human-Robot Interaction, pp. 241--248, (2008)
[14] Wobbrock, J. O., Wilson, A. D., and Li, Y. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In Proc. UIST 2007, 159-168.
[15] Suzuki, T., Ohya, A., Yuta, S.: Operation Direction to a Mobile Robot by Projection Lights. In: IEEE Workshop on Advanced Robotics and its Social Impacts, TPO-2-2, (2005)
[16] Accot J. and Zhai S. Refining Fitts' law models for bivariate pointing. Proceedings of ACM CHI 2003 Conference on Human Factors in Computing Systems, pp. 193–200 (2003).
[17] D.R. Olsen and M.A. Goodrich. Metrics for Evaluating Human-Robot Interactions, In Performance metrics for intelligent systems workshop, 2003
[18] MacKenzie I. S. and Buxton W. A. S. Extending Fitts' law to two-dimensional tasks. Proceedings of ACM CHI 1992 Conference on Human Factors in Computing Systems, pp. 219–226 (1992).
[19] Rubine, D. Specifying gestures by example. In Proc. SIGGRAPH 1991, 329-337.
[20] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology, volume 47, number 6, June 1954, pp. 381–391 (1954).
[21] S.Koceski, N. Koceska, I. Kocev, Design and Evaluation of Cell Phone Pointing Interface for Robot Control, International Journal of Advanced Robotic Systems (ISSN: 1729-8806. ISI Thomson Impact Factor: 0,375) (accepted paper)

[22] S.Koceski, N. Koceska, I. Kocev, Design and Evaluation of a Cell Phone Pointing Interface for Interaction with Large Projector Based Displays, International Journal of Computer Applications (ISSN: 0975 - 8887) (accepted paper)

[23] URI KARTOUN, HELMAN STERN, YAEL EDAN Virtual Reality Telerobotic System

[24] Raúl Marín, *Member, IEEE*, Pedro J. Sanz, *Member, IEEE*, P. Nebot, and R.Wirz

A Multimodal Interface to Control a Robot Arm via the Web: A Case Study on Remote Programming

[25] Terrence Fong Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation

[26] Zoe Doulgeri*, Member, IEEE,* and Tilemachos Matiakis*, Student Member, IEEE*

A Web Telerobotic System to Teach Industrial Robot Path Planning and Control

[27] Huosheng Hu, Lixiang Yu, Pui Wo Tsui, Quan Zhou Internet-based Robotic Systems for Teleoperation

[28] Igor R. Belousov 1, JiaCheng Tan2, Gordon J. Clapworthy 2 Teleoperation and Java3D Visualization of a Robot Manipulator over the World Wide Web

Димитрија Ангелков

Веб базирана далечинска контрола на мобилни роботи

УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП