

OŠ dr. Franja Žgeča Dornava
Dornava 136/a
2252 Dornava

Dornava, marec 2012

RAZISKOVALNA NALOGA

Avtorji:

Tomaž Majcen, učenec
Blaž Vidovič, dijak
Domen Tomše, dijak

Mentorja:

Danilo Muršec, učitelj
Franc Vrbančič, učitelj

OŠ dr. Franja Žgeča Dornava
Dornava 136/a
2252 Dornava

Ali lahko s telefonom zabijemo žebelj?

RAZISKOVALNA NALOGA

Avtorji:

Tomaž Majcen, učenec
Blaž Vidovič, dijak
Domen Tomše, dijak

Mentorja:

Danilo Muršec, učitelj
Franc Vrbančič, učitelj

Ptuj, marec 2012

Zahvala:

Zahvaljujemo se mentorjema za strokovne nasvete in pomoč pri pisanju poročila ter vsem, ki so kakorkoli pripomogli pri nastanku raziskovalne naloge.

Kazalo vsebine

1	Uvod.....	11
1.1	Opredelitev problema s trditvijo.....	11
1.2	Obrazložitev trditve	11
2	Fizikalni model zabijanja žebnja.....	12
3	Zgradba avtonomne naprave	14
3.1	Funkcionalni načrt sistema za iskanje in zabijanje žebnja	14
3.2	Strojni del.....	15
3.3	Montažni načrt.....	15
3.4	Delovanje robotskega sistema.....	15
3.5	Kratek opis eProDas-Rob1 učila in elementov.....	16
3.5.1	Električni načrt učila eProDas-Rob1	17
3.5.2	Pogon.....	17
3.5.3	Ostali elementi in uporabljen material.....	18
3.6	Kratek opis lego mindstorms NXT učila	19
3.6.1	Strojna oprema učila MindstromNXT	19
3.6.2	Programski ukazi učila Mindstorm NXT.....	19
3.7	Primerjava didaktičnih učil LEGO NXT in eProDas-Rob1.....	21
3.8	Programski del robota.....	22
3.8.1	Diagram poteka	22
3.8.2	Osnovni ukazi v programskem okolju Bascom	22
3.8.3	Splošni ukazi	23
3.8.4	Digitalni izhod	23
3.8.5	Ukaz Wait.....	23
3.8.6	Deklaracija spremenljivke.....	24
3.8.7	Zanka Do Loop	24
3.8.8	Programska zanka For-Next.....	24
3.8.9	Odločitveni stavek IF-Then Else	25

3.8.10	Analogni vhod.....	25
3.9	Empirični preizkus, rezultati preizkusa in vrednotenje rezultatov	27
3.9.1	Pravila preizkušanja robotskega sistema.....	27
3.9.2	Določitev maksimalnega časa iskanja žebnja	27
3.9.3	Opis preizkusnega poligona.....	28
3.9.4	Izmerjeni rezultati.....	28
3.9.5	Razprava o ugotovitvah empiričnega preizkusa	30
4	Dokazovanje dosegljivosti materialov in znanja	31
4.1	Vseprisotni materiali.....	31
4.2	Vseprisotno znanje.....	31
5	Končno vrednotenje trditve	33
6	Zaključek.....	34
7	Literatura in viri.....	35

Kazalo slik

Slika 1:	Funkcionalni načrt sistema za iskanje in zabijanje žebnja	14
Slika 2:	Montažni načrt.....	15
Slika 3:	Namestitev programskega okolja Bascom Basic.....	15
Slika 4:	Učilo eProDas_Rob1.....	16
Slika 5:	Električna shema eProDas–Rob1	17
Slika 6:	Servo motor	17
Slika 7:	Senzor svetlobe	
Slika 8:	Tipkalo (povečano x2)	18
Slika 9:	Lego kocke.....	18
Slika 10:	Lego Mindstorms NXT učilo s priključki	19
Slika 11:	Program učila lego NXT	20
Slika 12:	Diagram poteka delovanja robota	22
Slika 13:	Končna podoba robotskega sistema z lego NXT 2.0	26
Slika 14:	Glavni serviser Tomaž pri delu na robotskem sistemu	26

Slika 15: Risba preizkusnega poligona 28

Kazalo tabel

Tabela 1: Osnovni programski ukazi učila Mindstrom NXT 2.0.....	19
Tabela 2: Primerjava učil eProDas-Rob1 in lego NXT2.0	21
Tabela 3: Tabela izmerjenih rezultatov	28

Povzetek

Vsak človek se prej ko slej sreča z navidez preprosto nalogo zabiti žebelj. Šele z veliko vaje in ogromno časa, podana naloga dejansko postane dovolj enostavna, da preprosto vzamemo žebelj in ga s kladivom zabijemo, kjer ga rabimo.

Zabijanje žebnja je tudi predmet našega raziskovanja. V raziskovalni nalogi smo raziskovali ali je možno razviti in izvesti delujoč avtonomni sistem, ki zna poiskati in zabiti žebelj. Dodatno smo v raziskavi omejili, da smemo v sistem vgraditi le tiste elektronske dele in naprave, ki jih najdemo v vsakem domu ter žebelj obvezno zabiti s telefonom, cena sistema pa ne sme presegati nekaj deset evrov.

Za potrebe raziskave smo načrtovali in zgradili mobilni robotski sistem, katerega smo programirali za iskanje in zabijanje žebnjev. Z empirično raziskavo smo prikazali uspešnost sistema za izvedbo podane naloge.

Ključne besede: mikroprocesorski sistemi, program, algoritem, učilo

Summary

Every human being sooner or later comes across a seemingly simple task of nailing. Only with a lot of practice and a great amount of time the given task actually becomes simple enough to just take the nail and drive it with a hammer into the desired spot.

Nailing is also the subject of our research. In this research assignment, we conducted a research in order to determine whether it is possible to develop and execute a functional autonomous system that knows how to find and drive a nail. In addition, we limited the research by allowing only the electronic parts and devices, which can be found in every household, to be built into the system and to drive a nail only with a telephone, while the price of the system must not exceed more than several tens of Euros.

For the purpose of this research we projected and constructed a mobile robotic system, programmed for seeking and driving the nails. In the empirical research, we demonstrated the system's efficiency for executing the given task.

Keywords:

microprocessing, systems, program, algorithm, means of instruction.

Opredelitev uporabljenih strokovnih izrazov in besed

Strokovna beseda, kratica	Izpis, prevod	Opis
algoritem		Je grafično ali opisno zaporedje navodil (inštrukcij), ki rešijo podan problem. Algoritem je namenjen človeku. Uporablja se za lažje razumevanje in reševanje problema. Če algoritem (kodo) izvaja računalnik, potem govorimo o računalniškem programu.
avtonomno		samostojno
bascom basic		Programsko okolje v katerem s pomočjo programskega jezika bascom napišemo in preverimo program za didaktično učilo.
čip		Mini ploščica, običajno iz silicija, v ohišju in s priključki. Na površini ploščice je ogromno število elektronskih elementov. Namenjen je krmiljenju elektronskih vezij in procesov.
didaktično učilo		Zbirka procesorja, motorjev, senzorjev in ostalega materiala s pomočjo katerih lahko sestavimo in programiramo različne robotske sisteme.
eProDas-Rob1		Didaktično učilo razvito na Pedagoški Fakulteti, UL.
gravitacija		Pojav, da se mase privlačijo.
kladivo		Orodje nasajeno na ročaj in namenjeno zabijanju žebeljev ali tolčenje po materialu. Lahko tudi orodje pri določenih vrstah športa.
šala		Zapis ali govor, ki pri bralcu ali poslušalcu povzroča smeh. Npr. »Kaj je rjavo, stoji na travniku in se suče?« Krt pri metu kladiva. »Kaj je rjavo in skače po travniku?« Isti krt, ki mu je kladivo padlo na nogo.
lego Mindstorms NXT		Didaktično učilo, ki ga je razvilo podjetje Lego.
microcontroler	mikrokrmilnik	Mikrokrmilnik ali mikrokontroler je čip, ki vsebuje skoraj vse sestavine mikroročunalnika (procesor, notranji pomnilnik, vmesnike itd.). Za popoln mikroročunalnik mikrokrmilniku manjkajo le vhodno-izhodne enote, ki niso primerne za vgradnjo v čip. Mikrokrmilnik »živi in diha« s pomočjo programa, ki ga prenesemo v pomnilnik.
robot		Po definiciji Roberta Macniela iz leta 2009, je robot avtomatiziran mehanizem, ki je sposoben delno-samostojno ali v celoti samostojno opraviti določene naloge.
senzor		Komponenta strojne opreme, ki spreminja merjeno veličino v električno obliko.
telefon		V današnjem času precej razširjena naprava namenjena komuniciranju na daljavo. So različnih oblik, barv ...

trditev		Misel ali ideja, predvidoma izrečena v skladu z resničnostjo.
zabijanje		Postopek pri katerem s pomočjo kladiva povzročimo, da žebelj pride v material in tam običajno tudi ostane.
žebelj		Podolgovat predmet, namenjen pritrditvi, ki ima na enem koncu razširjeno glavo, na drugem pa je koničast za lažji prehod v materiale. Poznamo žebelje različnih oblik in dimenzij. Običajno so kovinski. Poznamo tudi lesene žebelje in žebelje iz drugih materialov.

1 Uvod

V začetku razvoja človeštva se je moral človek skozi življenje prebijati predvsem z močjo svojih mišic, roke, oči ..., skratka z uporabo tega, kaj mu je narava dala v obliki telesa oziroma je našel v naravi. Sčasoma je začel uporabljati orodja, najprej tista iz narave, kasneje se jih je naučil izdelovati tudi sam.

Orodjem so vedno hitreje sledila raznorazne naprave, vse do današnjih dni, ko z uporabo mini računalnikov in umetnih materialov znamo zgraditi marsikaj. Ne glede na kompleksnost zgrajenih naprav, pa človeško telo še vedno ostaja največji čudež narave, ki ga zaenkrat noben procesor ne more preseči. Kljub vsemu pa nam današnje naprave olajšajo marsikatero ročno opravilo.

1.1 Opredelitev problema s trditvijo

Zabijanje žebnja je navidez zelo enostavno opravilo, ko ga obvladamo. V resnici je potrebno kar nekaj let vaje in precej veščin, da žebelj zabijemo, brez da bi ga ukrivili oziroma brez da bi poškodovali prste s katerimi držimo žebelj. Lahko samo pritrdimo prizorom iz filmov in risank, kjer junaki, po seansi zabijanja žebnja, hodijo s povitimi prsti.

Trditev, ki jo bomo raziskali v okviru raziskovalne naloge lahko zapišemo v obliki:

T1: »Z dosegljivimi materiali in znanjem je možno narediti in programirati avtonomni sistem, ki najde in s telefonom zabije žebelj.«

1.2 Obrazložitev trditve

Pri dokazovanju trditve bomo uporabili metodo razčlenjevanja in našo kompleksno trditev razčlenili v delne trditve. Posamezno delno trditev bomo dokazovali s pomočjo študije obstoječih pisnih in elektronskih virov ter s pomočjo empiričnega preizkusa. Na osnovi rezultatov dokazovanja posameznih delnih trditve bomo sklepali na pravilnost osnovne trditve.

Če hočemo dokazati pravilnost trditve moramo dokazati besedne zveze in pojme, ki nastopajo v trditvi:

- dosegljivi materiali – dokazati, da so materiali iz katerega bi naredili sistem, dosegljivi,
- dosegljivo znanje – dokazati, da je znanje za sestavo in programiranje robota dosegljivo in razumljivo učencu/dijaku,
- avtonomnost sistema – ob pritisku na gumb start, sistem samostojno najde in zabije žebelj. Kakršen koli zunanji poseg ni dovoljen, drugače trenutni poskus šteje kot neuspešen. Merimo empirično zaganjanjem sistema in merjenjem uspešnosti iskanja in zabijanja žebnja v predvidenem času,
- narediti in programirati – dokazati, da je napravo možno narediti in programirati, da najde in s telefonom zabije žebelj.

Preden obrazložimo in dokažemo ali ovržemo posamezne dele trditve, se moramo seznaniti s fizikalnim dogajanjem procesa zabijanja žebnja in s strojno in programsko opremo naše naprave ter njenim delovanjem.

2 Fizikalni model zabijanja žebnja

Delo je v fiziki količina, ki meri prehajanje energije med telesi. Merimo ga v joulih – J; (1 J = 1 N m = 1 kg m²/s².) Delo sile v splošnem izračunamo kot integral sile po poti, pri čemer je delo A skalarna količina, rezultanta sil \mathbf{F} in pot s pa vektorski.

Po prvem zakonu termodinamike – zakonu o ohranitvi energije je sprememba energije sistema enaka vsoti prejetega in oddanega dela ter prejete in oddane toplote, pri čemer prejeta delo in toploto po dogovoru štejemo pozitivno, oddano pa negativno. Če je sistem toplotno izoliran od okolice, je torej delo enako spremembi energije sistema.

Energijo sistema sestavlja več vrst energije. Pri zabijanju žebnje se spreminjajo gravitacijska potencialna, kinetična in notranja. Kinetično energijo ima telo zaradi svojega gibanja. Ko telo z maso m rezultanta sil \mathbf{F} premakne za ds s hitrostjo \mathbf{v} , je delo dA , ki ga sila opravi in telo prejme, enako $dA = \mathbf{F} \cdot d\mathbf{s}$.

Z integracijo drugega Newtonovega zakona ugotovimo, da je to enako kinetični energiji $W_k = \frac{m \cdot v^2}{2}$.

Večji premik telesa sestavimo iz manjših, velja $A = \int \mathbf{F} \cdot d\mathbf{s} = W_{k2} - W_{k1}$, kjer je W_{k1} kinetična energija telesa v začetni točki, W_{k2} pa v končni točki.

Če deluje sila na telo v smeri gibanja, je delo pozitivno in kinetična energija telesa narašča. Z naraščanjem kinetične energije narašča tudi velikost hitrosti telesa. Če deluje sila v nasprotni smeri gibanja, je delo negativno in kinetična energija telesa pada. Ko pa je sila usmerjena pravokotno na smer gibanja, se kinetična energija ohranja, saj je delo enako nič.

Gravitacijsko potencialno energijo ima telo, na katerega drugo telo deluje z gravitacijsko silo. V resnici vsako telo deluje na druga telesa z to silo, izračunamo jo po formuli $F = \kappa \cdot \frac{m \cdot M}{r^2}$, kjer je F sila med telesoma, κ gravitacijska konstanta ($\kappa = 6,67 \cdot 10^{-11}$ N m²/kg²), M in m masi teles ter r razdalja med telesoma. Vendar so sile med telesi na Zemlji zelo majhne; gravitacijska sila med 10-gramskim žebnjem in 500-gramskim kladivom na razdalji 25 cm je npr. velikostnega reda 10^{-12} N. Ko obravnavamo energijo, jih lahko zato zanemarimo. Upoštevati pa moramo gravitacijsko silo med Zemljo in telesom, imenovano tudi teža telesa, ki jo lahko izračunamo po zgornji formuli ali ekvivalentno po drugem Newtonovem zakonu $F = m \cdot g$, kjer je F gravitacijska sila, m masa telesa in g težni pospešek.

Če na telo ne deluje nobena druga sila, je delo sile teže enako spremembi gravitacijske potencialne energije. Ko teža na telesu opravi delo (telo ga prejme), se mu poveča gravitacijska potencialna energija. Ta se lahko nato pretvori v drugo obliko energije, ali pa jo telo odda s tem, da samo opravi delo. Pri premiku telesa za ds lahko o delu sile teže povemo naslednje: $dA = \mathbf{F} \cdot d\mathbf{s} = m \cdot g \cdot ds = m \cdot g \cdot ds \cdot \cos\alpha = -m \cdot g \cdot dh$, pri čemer je α manjši kot med smerjo težnega pospeška in smerjo gibanja ter dh sprememba višine telesa, če višino merimo vzporedno s smerjo delovanja težnega pospeška. Delo gravitacijske sile pri premiku telesa s točke 1 na višini h_1 na točko 2 na višini h_2 je torej $A = m \cdot g \cdot (h_1 - h_2)$, ne glede na vodoravno razdaljo med točkama. Če je h_2 višje kot h_1 , se telesu pri premiku poveča gravitacijska potencialna energija za $m \cdot g \cdot (h_1 - h_2)$, če je h_2 nižje, pa zmanjša za toliko. Ničelno višino lege telesa lahko pri obravnavanju potencialne energije poljubno izberemo, a moramo vse višine določiti glede na to točko.

Ko utež prosto pada, se njena gravitacijska potencialna energija spreminja v kinetično. Če za ničelno višino izberemo vrh glave žablja ter zanemarimo zračni upor in trenje v vodilu, je tik pred udarcem po žablju utež na višini 0, njena potencialna energija je nič, njena kinetična energija pa takšna, kot je bila potencialna na začetni višini h : $W_{k2} = W_{p1} = m \cdot g \cdot h$, tu je W_{k2} kinetična energija na koncu, W_{p1} pa potencialna na začetku. Če po žablju udarimo s kladivom, pa ima kladivo poleg energije, ki je posledica spremembe njegove višine med udarcem, še dodatno kinetično energijo zaradi sile, s katero smo ga zavihteli. Tik pred dotikom žablja je energija kladiva enaka $W_k = \frac{m \cdot v^2}{2}$ kjer je v hitrost kladiva tik pred udarcem in je posledica tako našega vihtenja kot teže.

Ob trku z žabljem se večina te energije prenese na žebelj. Nekaj se je sprosti kot zvok, nekaj pa pretvori v notranjo energijo uteži/kladiva in žablja, zato se žebelj po več zaporednih udarcih nekoliko segreje. Kakšen delež energije se pretvori v kinetično energijo žablja je v veliki meri odvisno od materiala uteži oz. kladiva. Običajno jekleno kladivo prenese na žebelj okoli 70 % energije udarca, kladivo iz titana pa celo do 97 % (po podatkih proizvajalca titanovih kladiv Stiletto Tool Company). Žebelj potrebuje enako energije, da se zabije v les, a z boljšim kladivom je za enak učinek potrebnega manj dela z naše strani.

Opomba: Fizikalno dogajanje pri zabijanju žablja smo prevzeli po literaturi [6]

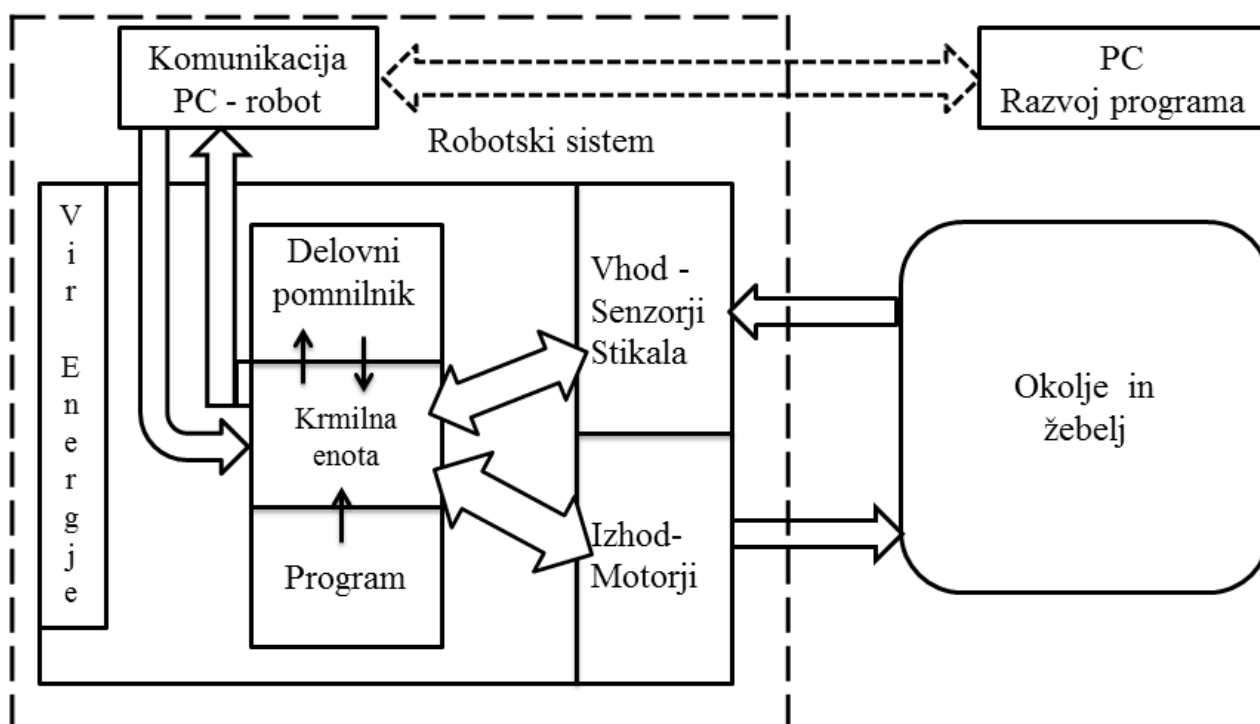
3 Zgradba avtonomne naprave

Za potrebe raziskovalne naloge smo načrtovali in izvedli strojni in programski del avtonomnega robota, ki išče in zabija žeblje. Odločili smo se, da bomo zgradili robota z dvema različnima pristopoma. Prvega robota smo sestavili in programirali s pomočjo didaktičnega učila Lego NXT, ki v obliki kompleta vsebuje potreben material in programsko opremo za programiranje.

Pri sestavi drugega robota smo si pomagali z didaktičnim učilom eProDas-Rob1, ki je bilo razvito na Pedagoški Fakulteti, Univerze v Ljubljani pod vodstvom dr. Slavka Kocijančiča. Pri tem učilu je potrebno več znanja tako pri programiranju kot pri sestavi robota. V nadaljevanju bomo opisali zgradbo robota in programiranje le tega, tako z enim kot z drugim učilom. Učili bomo tudi primerjali po izbranih kriterijih kot so cena, dosegljivost navodil, enostavnost programiranja, popularnost in enostavnost konstrukcije ogrodja.

3.1 Funkcionalni načrt sistema za iskanje in zabijanje žeblja

Na sliki 1 je predstavljen funkcionalni načrt robotskega sistema ter interakcija tega sistema z okoljem.



Slika 1: Funkcionalni načrt sistema za iskanje in zabijanje žeblja

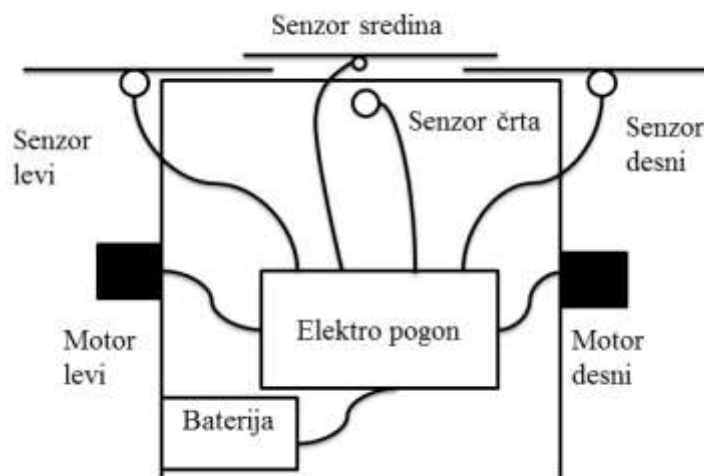
3.2 Strojni del

Strojni del robotskega sistema je sestavljen iz:

- senzorjev s pomočjo katerih zaznavamo robove mize in lego robota,
- motorjev s pomočjo katerih robota premikamo in zabijamo žebelje,
- krmilne enote, ki procesira informacije s senzorjev in stikal ter ustrezno krmili delovanje motorjev,
- vira energije,
- USB priključka preko katerega na robota prenesemo program, ki ga običajno razvijamo na osebнем računalniku.

3.3 Montažni načrt

Na sliki 2 je prikazan montažni načrt strojnih elementov robota.



Slika 2: Montažni načrt

3.4 Delovanje robotskega sistema

Robot se giblje na beli površini, ki je omejena s črno črto. Ob pritisku na gumb start se robot začne pomikati naprej, dokler ne najde žebelja ali naleti na črno črto. Zaznavanje črne črte poteka s pomočjo svetlobnega senzorja na spodnji strani robota (senzor črta), zaznavanje žebelja pa preko stikal na prednji strani robota (senzorji levo, desno in sredina).

Če robot najde črno črto, se ustavi, nato se pomakne nazaj za nekaj centimetrov in se spet ustavi. Sledi rotacija robota za določen kot levo ali desno, kar je odvisno od algoritma izogibanja črte. Po končani rotaciji se začne robot spet pomikati naprej.

Če robot zazna žebelj z levim ali desnim stikalom, se ustavi. Nato se pomakne nazaj za nekaj centimetrov in se spet ustavi. Sledi rotacija robota za določen kot levo ali desno, z željo dobiti žebelj v sredinsko lego, kjer ga bomo lahko zabili. Po končani rotaciji, se robot pomakne naprej in preko stikal ponovno preveri lego žebelja. Po potrebi robot opisan postopek ponavlja, dokler žebelja ni v sredinski legi.

Če robot zazna žebelj s sredinskim stikalom, se ustavi in sproži postopek zabijanja žebelja. Ko je žebelj zabiti, robot ustavi vse motorje in čaka na ponovni start ukaz.

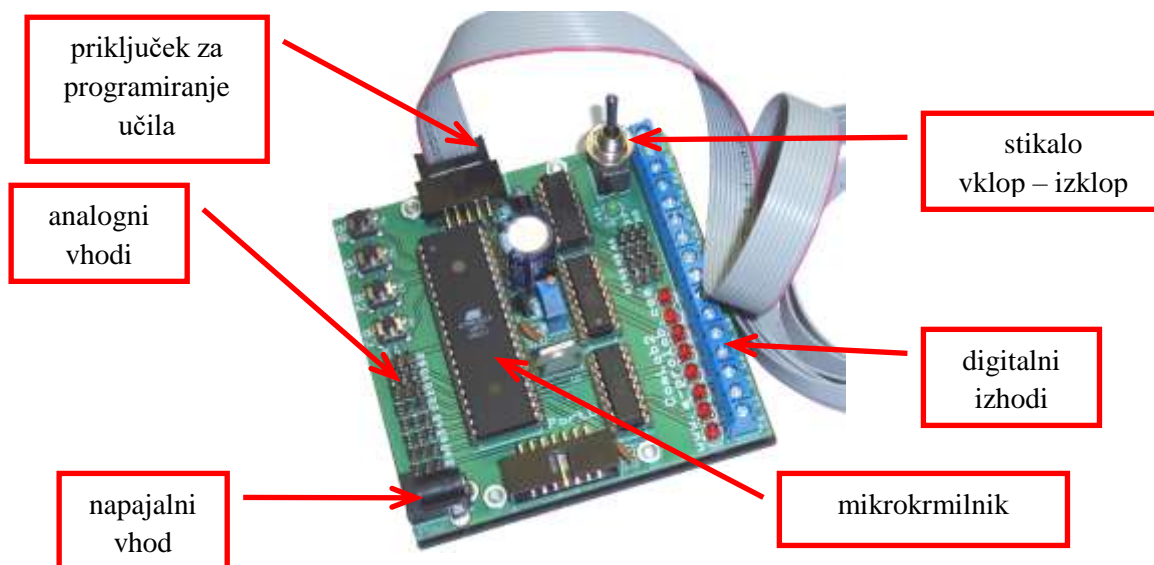
3.5 Kratek opis eProDas-Rob1 učila in elementov

EProDas-Rob1 učilo se lahko programira z različnimi programskimi orodji: Bascom basic, C++, Delphi, Visual Basic in LabView. Mi smo ga programirali z brezplačno verzijo programa Bascom basic, ki je dosegljiva na spletni strani <http://www.mcselec.com>. Omejitev brezplačne različice je, da prevedena koda ne sme presegati velikosti 4k byte, kar je popolnoma zadostovalo našim potrebam.

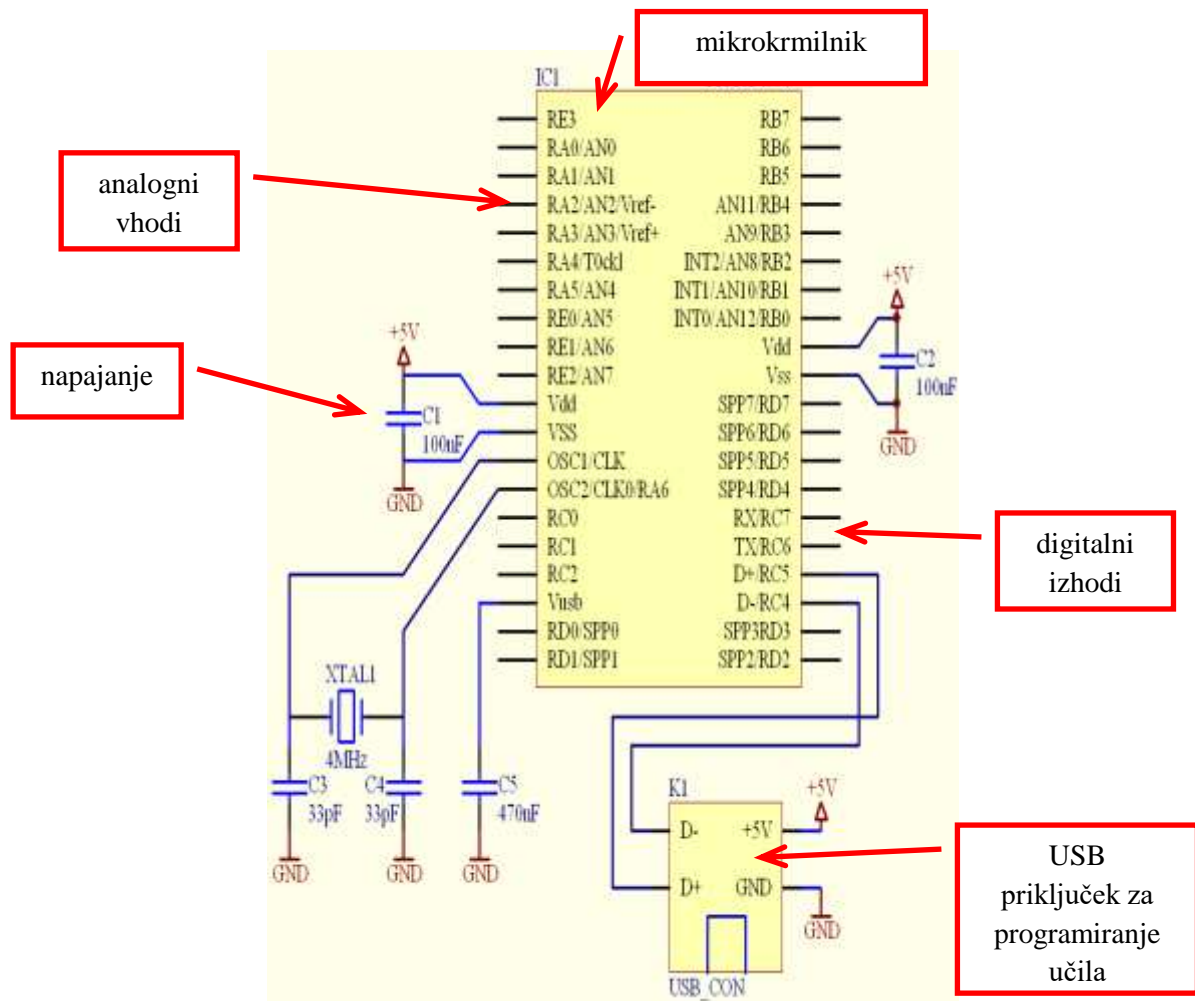
Namestitev programa je zelo enostavna. Z zgoraj navedene spletne strani smo presneli instalacijsko datoteko, jo zagnali in sledili navodilom instalacije. (slika) Če ne spreminjamo namestitve, se program namesti na disku C v imeniku MSC Electronics. Zagon programa je možen preko bližnjice na namizju ali preko ukazov `start/vsi_programi/ MSC_Electronics/Bascom_AVR`.



Srce strojnega dela učila je mikrokrmilnik Atmega 16, ki ima 32 priključkov, ki jih lahko programiramo kot vhode ali izhode. Učilo ima tudi 8 analogno digitalnih pretvornikov preko katerih lahko merimo veličine kot so napetost, temperatura ali svetloba odvisno od senzorja, ki je priključen a na analogni vhod. Napajanje učila je možno preko adapterja (7,5 V) ali preko baterij.



3.5.1 Električni načrt učila eProDas-Rob1



Slika 5: Električna shema eProDas-Rob1

(vir :lie.fe.uni-lj.si)

3.5.2 Pogon



Živa bitja za premikanje po prostoru uporabljajo mišice. Mišice skupaj s kostmi in drugimi organi omogočajo gibanje živemu bitju. Za premikanje robota smo morali najti ustrezni mehatronski nadomestek. Po raziskavi ponudb na tovrstnem področju smo se odločili, da bodo za naš namen najprimernejši servo motorji.

Slika 6: Servo motor

(vir: www.c2.com/cybords)

3.5.3 Ostali elementi in uporabljen material

Stikala smo uporabili za zaznavanje lege žebnja glede na sredinsko lego robotskega sistema. Žebelj smo zaznavali s pomočjo tipkal. Energijo za delovanje je robotski sistem dobival iz baterije. Za zaznavanje črne črte na beli površini smo uporabili svetlobni senzor. Ohišje robota smo zgradili iz lego kock.

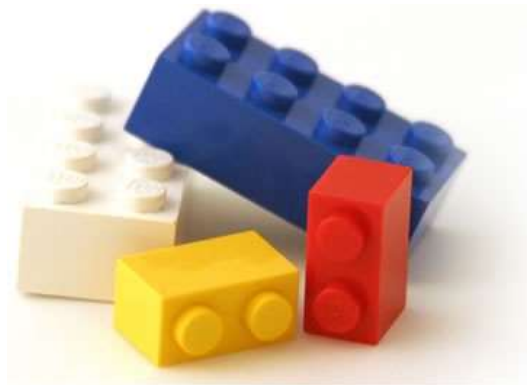


Slika 7: Senzor svetlobe
(vir: www.sparkfun.com)



Slika 8: Tipkalo (povečano x2)
(vir: www.ena.com)

Za ohišje robota smo uporabili lego kocke.



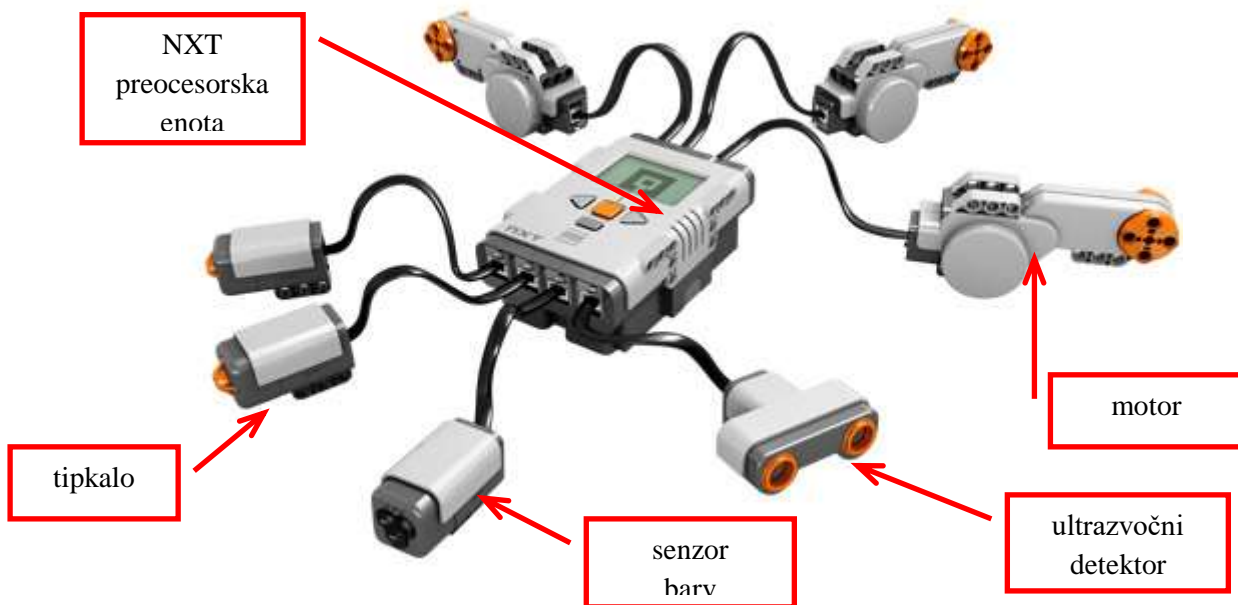
Slika 9: Lego kocke
(vir: www.lifeclass.net)

3.6 Kratek opis lego mindstorms NXT učila

Učilo lahko programiramo s programskimi orodji NXT-G, LabView in NXC.

3.6.1 Strojna oprema učila MindstromNXT



Jedro sistema je procesorska enota NXT, na katero priključimo stikala, senzorje motorje in podobne elemente kot je to razvidno iz slike 10. Po želji lahko dodatne elemente dokupimo.

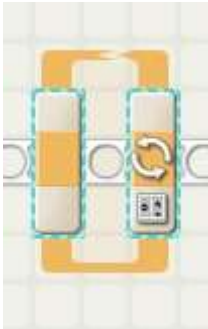
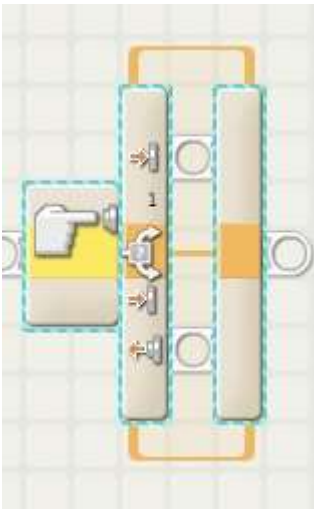


Slika 10: Lego Mindstorms NXT učilo s priključki
(vir: mindstorms.lego.com)

3.6.2 Programski ukazi učila Mindstorm NXT

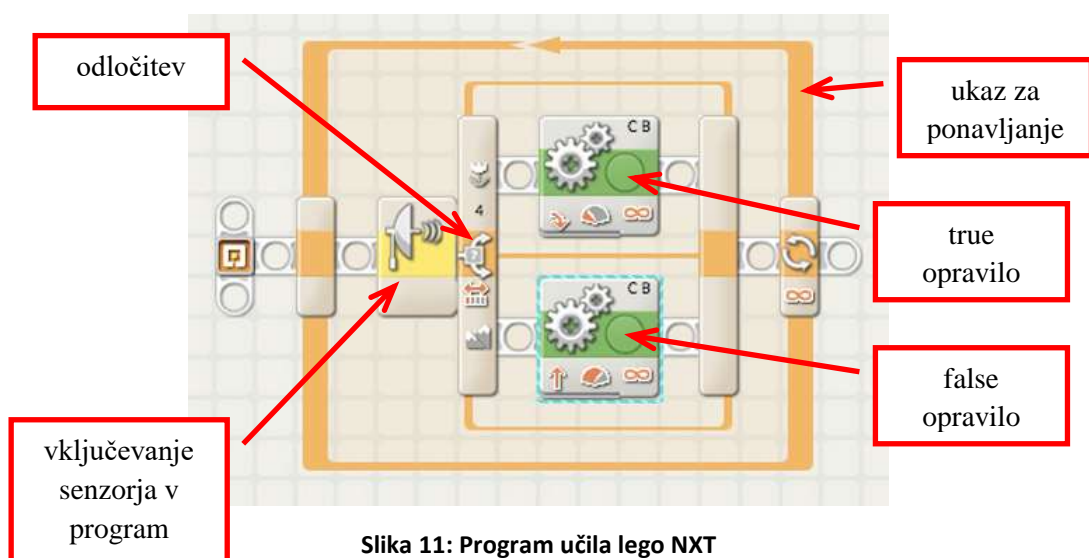
Tabela 1: Osnovni programski ukazi učila Mindstrom NXT 2.0

NXT-G	NXC	Pomen
	OnFwd(OUT_A, 75);	vklop motorja A naprej
	Off(OUT_BC);	Ustavi motorja C in B

NXT-G	NXC	Pomen
	<pre>repeat(4) { Opravilo; }</pre>	<p>Ukaz za ponavljanje - Opravilo ponovi 4 krat.</p>
	<pre>if(pogoj) { Opravilo A; }else { Opravilo B; }</pre>	<p>Ukaz za odločanje (If stavek) Če je pogoj resničen (true), izvedi opravilo A drugače izvedi opravilo B.</p>

3.6.2.1 Primer programa

Na sliki 11 je primer programa, kjer z ultrazvočnim senzorjem zaznavamo ovire. Če ni ovire peljemo naprej, drugače zavijamo in oviro obvozimo.



Slika 11: Program učila lego NXT

3.7 Primerjava didaktičnih učil LEGO NXT in eProDas-Rob1

Učila smo primerjali po kriterijih kot so cena, težavnost programiranja, možnost nadgradnje, dostopnost učila in podobnih.

Tabela 2: Primerjava učil eProDas-Rob1 in lego NXT2.0

Kriteriji	Učilo	eProDas-Rob1	Lego NXT 2.0
Cena procesorske enote v evrih		30	213
Programiranje		Bascom basic, C++, Delphi, vse oblike Microsoft Visual orodij in LabView.	NXT-G, LabView, NXC.
Nadgradnja – dodatni senzorji, stikala in motorji		32 vhodov, ki so lahko tudi izhodi. Na vhod priključimo stikalo ali senzor na izhod pa motorje.	Štirje izhodi in štirje vhodi. Nadgradnja je mogoča z nakupom dodatne NXT procesorske enote, ki stane 213 evrov.
Konstrukcija izdelka		Poljubna, lahko tudi z lego kockami.	Poljubna z uporabo lego kock.
Dostopnost navodil		Prevedeno v 12 evropskih jezikov: slovenščina, angleščina, nemščina, nizozemščina ...	Le v angleščini.
Popularnost		Velika med poznavalci, manjša med ostalimi.	Velika, dokler plačujejo starši. Potem se raje odločijo za cenejši sistem recimo eProDas-Rob1.
Priporočamo		Začetnikom in izkušenim uporabnikom.	Začetnikom in entuziastom.

Učilo eProDas-Rob1 bi predlagali za dijake in za izkušenejšše učence v višjih razredih (zadnja triada) in za tiste, ki bodo nadaljnje učne poti vodile na področje elektrotehnike, mehatronike ali računalništva. Ni treba posebej omenjati, da mora tudi učitelj-mentor iti dovolj več del s tem učilom.

Učilo Lego Mindstorm 2.0 bi predlagali predvsem učencem nižjih letnikov in tistim, ki se prvič srečajo s področji tehnike. Največja slabost učila je cena in dejstvo, da ga ni moč nadgraditi z deli, iz drugih sestavljanj ali z deli, ki bi jih kupili v tehničnih trgovinah. Enkrat Lego vedno Lego.

Analizo obeh učil lahko zaključimo z besedami, da bi bilo za poučevanje dobro imeti obe učili in to v zadostni količini.

3.8.3 Splošni ukazi

```
$regfile = "m16def.dat"
```

```
$crystal = 8000000
```

V vsaki vrstici lahko zapišemo komentar, ki nam pove kaj bo ta vrstica v programu izvedla označimo jo takole:

```
'Moj prvi komentar
```

Vsak program na koncu zaključimo z End

```
End 'end program
```

V vsakem novem programu najprej določimo s čem bo program komuniciral, vhodi ali izhodi in to zapišemo kot je razvidno iz spodnjih vrstic.

'Inicializacija priključkov na vhod ali izhod

```
Config PORTA = Input /Output
```

```
Config PORTB = Input
```

```
Config PORTC = Output
```

```
Config PORTD = Input /Output
```

3.8.4 Digitalni izhod

'Uporaba priključkov tipa izhod

```
PORTC = xxx
```

```
PORTC = &B11111111
```

```
PORTC.0 = x
```

Prva možnost je ukaz **PORTC = *stevilo*** (3). V parameter *stevilo* vpišemo desetiško število tipa byte, torej število med 0 in 255, ki ga želimo prikazati na izhodu. Lahko pa to zapišemo tudi drugače, in sicer z ukazom **PORTC = &B11111111** (2). Črka B pomeni, da bomo število zapisali v binarni obliki (dvojiški sistem). Osem enic pomeni, da je vseh 8 bitov v logični enici. S tem ukazom pravzaprav lahko nastavljamo vsak bit posebej. Na primer ukaz **PORTC = &B11110000** pomeni, da bodo biti od 0-3 v logični "0", višji štirje biti pa v logični "1". V tem primeru zapišemo **PORTC.0 = *logično stanje*** V parameter *logično stanje* zapišemo 1, če želimo izhod postaviti v logično "1", ali 0, če želimo izhod postaviti v logično "0".

3.8.5 Ukaz Wait

```
Wait 1
```

```
Waitms 100
```

```
Waitus 1000
```

V programu smo uporabili ukaz **Wait** 1, s katerim smo prekinili izvajanje programa za 1s. Število za ukazom **Wait** pomeni število sekund. Če želimo čakati manj, lahko uporabimo ukaza **Waitms** ali **Waitus**. V prvem primeru zapišemo časovni interval v mili sekundah, v drugem primeru pa časovni interval v mikrosekundah.

3.8.6 Deklaracija spremenljivke

Dim Stevec **As Byte**

Novo spremenljivko **stevec** smo deklarirali kot spremenljivko tipa **byte**, za kar smo uporabili ukaz **Dim** ime_spremenljivke **As Byte**. Na ta način lahko deklariramo še druge spremenljivke, ki pa so lahko tipa **byte**, **integer**, **word**, **long**, **single**, **double**, **string** itd.

3.8.7 Zanka Do Loop

Do

Ukaz 1

Ukaz 2

...

Loop

Navadno pri avtonomnih sistemih želimo, da se neko zaporedje ukazov nenehno ponavlja. Na voljo imamo več možnosti in ena izmed njih je zanka **Do-loop**. Vsi ukazi, ki jih vpišemo med ti dve besedi, se bodo nenehno ponavljali, vse dokler iz zanke ne izstopimo.

Do

Ukaz 1

Ukaz 2

...

Loop Until pogoj

Zgoraj smo videli, da je zanka do-loop pravzaprav neskončna zanka. Lahko pa ji dodamo pogoj, s katerim določimo, kdaj naj se izvajanje konča. Namesto ukaza loop zapišemo ukaz **Loop Until** pogoj (7). Torej, zanka se bo ponavljala vse dokler ne bomo izpolnili pogoja. Slednji je lahko poljuben. V našem primeru smo dodali števec, ki se vsakokrat poveča za 1. Ko je vrednost števca večja od 14, se izvajanje zanke konča.

3.8.8 Programska zanka For-Next

For I = zacetna_vrednost **to** koncna_vrednost

Ukaz 1

Ukaz 2

...

Next I

Kot smo že omenili, ponavljanje programa lahko dosežemo na več načinov. Eden od njih je tudi uporaba zanke **For-Next**. Lastnost te zanke je, da lahko določimo tudi začetno vrednost števca, katero vpišemo v parameter `zacetna_vrednost`. Zanka se bo ponavljala dokler stevec ne doseže končne vrednosti, katero določimo s parametrom `koncna_vrednost`.

3.8.9 Odločitveni stavek IF-Then Else

If pogoj Then

Ukaz 1

Else

Ukaz 2

End If

Če je pogoj izpolnjen (true) izvršimo ukaze med **Then** in **Else** drugače izvršimo ukaze med **Else** in **End If**. Else blok ukazov lahko po potrebi tudi izpustimo. Če else blok ukazov izpustimo, potem odločitveni stavek zapišemo kot:

If pogoj Then

Ukaz 1

End If

Običajno v kombinaciji z digitalnim vhodom uporabljamo pogojne stavke **if**. Torej, če je nek pogoj (na primer stanje na digitalnem vhodu) izpolnjen, naj se izvrši določen ukaz. Lahko napišemo tudi več ukazov, in sicer enega pod drugim Vsak pogojni stavek moramo zaključiti z besedo **End if**. Lahko pa pogojni stavek zapišemo v eni vrstici, in v tem primeru ukaz **End If** odpade. Pri tako zapisanem pogojnem stavku moramo ukaz zapisati v isti vrstici:

```
If pogoj Then Ukaz 1
```

3.8.10 Analogni vhod

```
Config ADC = Single , Prescaler = Auto , Reference = Off 'inicializacija AD pretvornika
```

```
Start ADC 'zagon delovanja vezja
```

```
Napetost = Getadc(6) 'čitanje vrednosti s 6. vhoda
```

Poleg digitalnih vhodov, lahko z vmesnikom eProDas-Rob1 spremljamo osem 10-bitnih analognih vhodov. Ukaz za konfiguracijo ADC-ja je zapisan zgoraj. Z nastavitvijo parametra `Prescaler` na vrednost `Auto`, nastavimo največjo frekvenco vzorčenja, z nastavitvijo parametra `Reference` na `off` pa povemo, da imamo zunanjo napetostno referenco. Z ukazom `start ADC` zaženemo analogni digitalni pretvornik .

Napetost na vhodu preberemo z ukazom **Getadc**, ki nam vrne vrednost tipa word in lahko zavzame vrednosti od 0-1023. V oklepaj zapišemo oznako analognega vhoda, ki ga želimo spremljati ,od 0-7 . V BASCOM-u lahko uporabljamo različne operatorje. V pogojnih stavkih tako pogosto naletimo na operatorje; različno ($<>$), manjše ($<$), večje ($>$), enako ($=$), manjše ali enako ($<=$), in operator večje ali enako ($>=$) . Uporabljamo lahko tudi logična operatorja or (ali) ter and (in).



Slika 13: Končna podoba robotskega sistema z lego NXT 2.0



Slika 14: Glavni serviser Tomaž pri delu na robotskem sistemu

3.9 Empirični preizkus, rezultati preizkusa in vrednotenje rezultatov

Pred začetkom izvajanja preizkusa delovanja robotskega sistema smo določili smiselna pravila, ki smo se jih držali ves čas preizkušanja naprave. Pred začetkom izvajanja 100 preizkusov, smo izvedli deset testnih preizkusov, s katerimi smo umerili delovanje naprave. V teh desetih preizkusih smo empirično določili nekatere parametre gibanja robota kot so: hitrost gibanja, kot rotacije, dolžino gibanja nazaj in podobne.

3.9.1 Pravila preizkušanja robotskega sistema

Pravila veljavna pri empiričnem preizkušanju delovanja avtonomne robotske naprave:

- Velikost iskalne površine: 2 x 1 meter-bele barve;
- 10 cm od roba črna obroba. Obrobo robot zazna in potem ne pade z mize;
- Izhodiščna točka robota: Robot začne iskanje na eni od 8. točk – pozicijo žrebamo pred vsakim poskusom;
- Naloga: robot naključno poišče in zabije žebelj, ki je postavljen v eno izmed 6 možnih pozicij. Pozicijo žeblja za vsak poskus izberemo z metom kocke;
- Pred začetkom preizkusa 6 naključno izbranih učencev določi pozicije žeblja in 8 naključno izbranih učencev izbere startne pozicije robotskega sistema;
- Število ponovitev: 100 poskusov, čas izvajanja posameznega poizkusa je do 3 min. Vsak poskus, ki traja dlje bomo označili kot neuspešen;
- Med preizkušanjem, se ne sme posegati v programski ali strojni del robotske naprave. Dovoljeno je menjava baterije in pritisk na gumb start;
- Med preizkušanjem je treba poskrbeti, da so zunanji pogoji kot so osvetljenost prostora, število ljudi okrog mize, lega miza v prostoru in podobno, kar se le da konstantni in enaki za vsak izmed sto preizkusov.
- Izvajalci preizkusa: Tomaž Majcen s pomočniki.

3.9.2 Določitev maksimalnega časa iskanja žeblja

Čas iskanja žeblja smo omejili na 3 minute. Čas smo empirično izmerili na desetih prostovoljcih starih med 12 in 75 let – po dva prostovoljca iz posamezne starostne skupine:

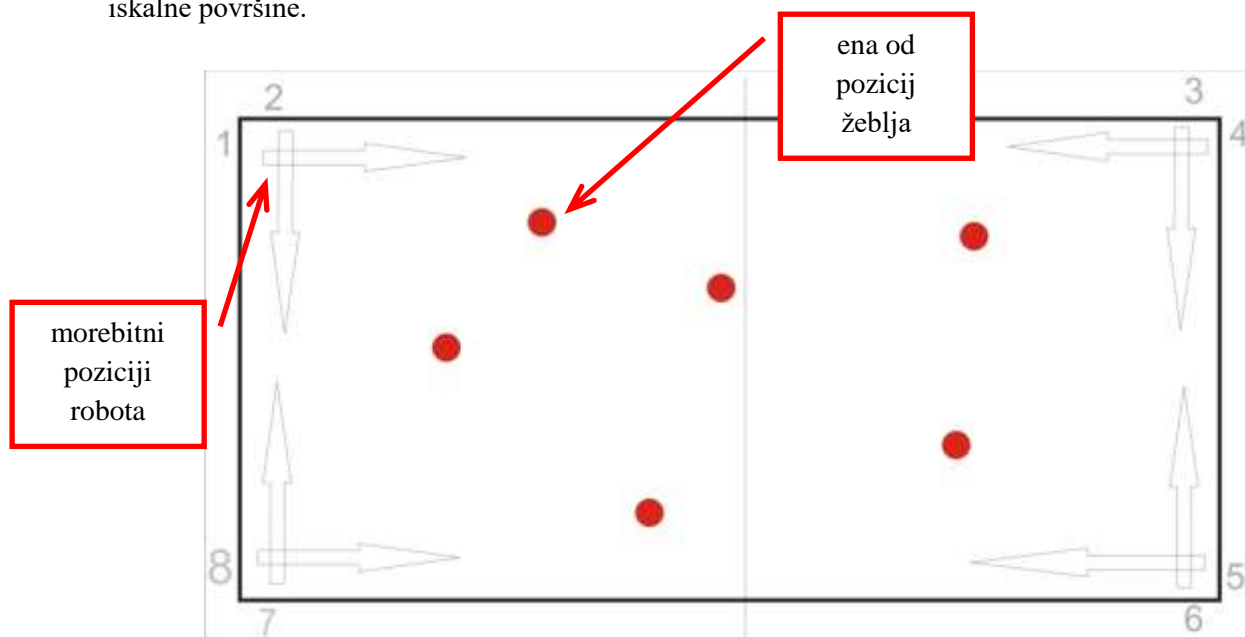
- od 12 do vključno 15 let – zadnja triada OŠ,
- nad 15 let do vključno 19 let – srednja šola,
- nad 19 let do vključno 23 let – višja in visoka šola,
- nad 23 let do vključno 65 let – aktivno obdobje ter
- nad 65 let – tretje življenjsko obdobje.

Za vsakega prostovoljca smo izmerili čas, ki ga je porabil, da je s 200 g. kladivom zabil 20 centimetrski žebelj v lesen štor. Časi zabijanja so se gibali med 2 in pol minute ter tri in pol minute. Na osnovi časovnega intervala v katerem človek zabije žebelj smo se odločili, da mora robotski sistem najti in zabiti žebelj v 3 minutah ali pa bomo poskus označili kot neuspešen.

Empirični preizkus delovanja robota bomo opravili le s sistemom, ki je bi zgrajen s pomočjo didaktičnega kompleta Lego NXT.

3.9.3 Opis preizkusnega poligona

Velikost in barva površine po kateri robotski sistem išče žebelj je opisan v pravilih izvajanja empiričnega preizkusa iskanja žebelja (glej začetek tega poglavja). V teh pravilih smo opredelili tudi ostale elemente iskalne površine.



Slika 15: Risba preizkusnega poligona

Pred začetkom preizkusa žebelj naključno postavimo v eno izmed šestih pozicij (rdeča pika). Robota naključno postavimo in usmerimo v smeri puščice v eno izmed osmih startnih pozicij (siva števila in puščice).

3.9.4 Izmerjeni rezultati

Tabela 3: Tabela izmerjenih rezultatov

Zap. štev. preizkusa	Št. začetne smeri	Št. pozicije žebelja	Čas izvajanja naloge v (s)	Opombe
1.	6	4	26	poskus uspešen
2.	4	1	70	poskus uspešen
3.	5	1	88	poskus uspešen
4.	6	2	18	poskus uspešen
5.	8	4	12	poskus uspešen
6.	2	6	7	poskus uspešen
7.	5	1	65	poskus uspešen
8.	2	5	108	poskus uspešen
9.	1	1	10	poskus uspešen
10.	7	1	93	poskus uspešen
11.	1	1	12	poskus uspešen
12.	4	3	X	poskus neuspešen
13.	4	2	X	poskus neuspešen
14.	3	2	11	poskus uspešen
15.	4	2	X	poskus neuspešen
16.	3	5	5	poskus uspešen
17.	3	4	10	poskus uspešen

Zap. šte. preizkusa	Št. začetne smeri	Št. pozicije žeblja	Čas izvajanja naloge v (s)	Opombe
18.	4	6	11	poskus uspešen
19.	8	6	93	poskus uspešen
20.	1	6	46	poskus uspešen
21.	5	5	54	poskus uspešen
22.	4	2	24	poskus uspešen
23.	3	6	25	poskus uspešen
24.	2	1	118	poskus uspešen
25.	2	1	53	poskus uspešen
26.	6	1	110	poskus uspešen
27.	2	1	52	poskus uspešen
28.	8	6	49	poskus uspešen
29.	7	1	32	poskus uspešen
30.	8	3	8	poskus uspešen
31.	8	4	34	poskus uspešen
32.	3	6	30	poskus uspešen
33.	1	6	8	poskus uspešen
34.	1	6	62	poskus uspešen
35.	7	2	12	poskus uspešen
36.	3	1	22	poskus uspešen
37.	4	3	20	poskus uspešen
38.	5	6	27	poskus uspešen
39.	1	4	11	poskus uspešen
40.	8	4	9	poskus uspešen
41.	1	3	32	poskus uspešen
42.	8	6	64	poskus uspešen
43.	5	5	17	poskus uspešen
44.	6	2	48	poskus uspešen
45.	7	1	27	poskus uspešen
46.	8	3	8	poskus uspešen
47.	6	2	47	poskus uspešen
48.	3	1	X	poskus neuspešen
49.	7	3	X	poskus neuspešen
50.	4	5	47	poskus uspešen
51.	5	6	40	poskus uspešen
52.	6	1	119	poskus uspešen
53.	3	4	16	poskus uspešen
54.	8	5	14	poskus uspešen
55.	6	5	14	poskus uspešen
56.	4	2	67	poskus uspešen
57.	5	4	4	poskus uspešen
58.	1	1	X	poskus neuspešen
59.	7	1	X	poskus neuspešen
60.	6	3	73	poskus uspešen
61.	2	6	4	poskus uspešen
62.	4	2	66	poskus uspešen
63.	4	2	32	poskus uspešen
64.	4	6	26	poskus uspešen

Zap. št. preizkusa	Št. začetne smeri	Št. pozicije žebnja	Čas izvajanja naloge v (s)	Opombe
65.	6	2	18	poskus uspešen
66.	5	5	7	poskus uspešen
67.	6	1	15	poskus uspešen
68.	2	2	22	poskus uspešen
69.	8	1	58	poskus uspešen
70.	1	4	52	poskus uspešen
71.	8	1	49	poskus uspešen
72.	3	2	45	poskus uspešen
73.	1	4	34	poskus uspešen
74.	2	3	25	poskus uspešen
75.	7	3	18	poskus uspešen
76.	4	4	6	poskus uspešen
77.	6	1	8	poskus uspešen
78.	8	2	45	poskus uspešen
79.	7	6	26	poskus uspešen
80.	3	6	105	poskus uspešen
81.	3	4	12	poskus uspešen
82.	4	2	22	poskus uspešen
83.	4	6	37	poskus uspešen
84.	2	1	55	poskus uspešen
85.	7	4	12	poskus uspešen
86.	8	2	100	poskus uspešen
87.	8	1	65	poskus uspešen
88.	5	5	30	poskus uspešen
89.	4	4	42	poskus uspešen
90.	8	2	19	poskus uspešen
91.	7	2	11	poskus uspešen
92.	2	6	7	poskus uspešen
93.	3	5	4	poskus uspešen
94.	5	1	44	poskus uspešen
95.	3	6	80	poskus uspešen
96.	7	3	54	poskus uspešen
97.	8	1	10	poskus uspešen
98.	6	4	20	poskus uspešen
99.	4	1	16	poskus uspešen
100.	2	1	32	poskus uspešen

3.9.5 Razprava o ugotovitvah empiričnega preizkusa

Robotski sistem je bil neuspešen v sedmih preizkusih. Pri vseh neuspešnih preizkusih je do neuspešnosti prišlo pri obračanju robotskega sistema, ko izvajal manever izogibanja črni črti. Pri obračanju se žebelj bočno zataknil med senzorje in ostalo konstrukcijo sistema. Ker je šlo za konstrukcijsko napako, ki jo je enostavno odpraviti (z bočnimi senzorji) in ker je robot kljub vsemu bil uspešen v 93 % primerov, smo se soglasno odločili, da lahko delovanje robota označimo kot uspešno.

Konstrukcijske napake nismo odpravili, saj bi to zahtevalo nakup dodatne NXT procesorske komponente, ki stane dobrih 200 evrov. Nakup dodatne NXT procesorske komponente bi bil potreben, ker so na posamezni komponenti le štirje vhodi za senzorje in stikala. Glede na to, da strošek ne bi bil smiseln, smo se odločili, da nakupa ne opravimo.

4 Dokazovanje dosegljivosti materialov in znanja

Pri dokazovanju trditve smo uporabili metodo razčlenjevanja in našo kompleksno trditev razčlenili v delne trditve. Posamezne delne trditve smo že dokazali. Dokazati moramo še dosegljivost materialov ter dosegljivost in razumljivost znanja, ki so potrebni za izgradnjo robotskega sistema, kar bomo dokazovali s pomočjo študije obstoječih pisnih in elektronskih virov.

4.1 Vseprisotni materiali

Naš sistem je sestavljen iz:

- Ogrodje naprave je iz lego kock. Preverili smo med sošolci in ugotovili, da med 30 sošolci ni sošolca, ki ne bi imel lego kock doma oziroma, ki ne bi imel poznal vsaj enega prijatelja, ki ima lego kocke. Iz tega sklepamo, da so lego kocke splošno dosegljive. V nekaterih trgovinah jih imajo celo toliko, da jih prodajajo, kar samo dodatno potrjuje pravilnost našega sklepa o splošni dosegljivosti materiala za ogrodje naprave;
- Za delovanje naprave smo potrebovali motorčka, baterijo, procesorsko enoto, žice in podoben elektronski material. Naštet material je enostavno in poceni dosegljiv v specializiranih trgovinah kot npr. HTE Maribor. Na spletu smo tudi našli podatke, da je v vseh večjih slovenskih mestih, razen Ptuja, povprečno vsaj ena podobno specializirana trgovina z elektronskim materialom kot: baterije, motorčki, omrežni adapterji, stikala, tipkala, diode in žice. Iz tega sklepamo, da potrebni elektronski materiali in elementi, ki jih potrebujemo za napravo, splošno dosegljivi.

Glede na zapisano lahko potrdimo delni trditvi, da je v Sloveniji več kot dovolj možnosti priti poceni in enostavno do vsega potrebnega materiala za izgradnjo naprave, kot smo si jo zamislili.

4.2 Vseprisotno znanje

Če hoče učenec sestaviti delujočo napravo, mora poznati in razumeti:

- osnovne postopke sestavljanja kot so združevanje, podaljševanje, gibljivi sklop in podobne,
- osnovne gradnike elektronskih vezij (motor, baterija, žica, upor, stikalo) in njihovo delovanje,
- osnove programiranja.

Iz javno veljavnih podatkov:

- literatura [1, 3, 4, 5, 9]
- • eden izmed ciljev javnih izobraževalnih programov je tudi: "učenci/dijaki pridobijo znanja za uspešno rabo modernih komunikacijskih tehnologij za delo na strokovnem področju", ki se pojavlja v učnih načrtih za osnovno in srednjo šolo pri modulih in predmetih kot so: tehnika in tehnologija, računalništvo in dokumentacija, informacijski sistemi, informatika, knjižnična informacijska znanja in drugih. [4] Še več EU je ta cilj vnesla med kot temeljno smernico vseživljenjskega izobraževanja,
- • cilj je zapisan tudi v [7],
- • Evropska komisija za izobraževanje in usposabljanje na svojih spletnih straneh omenja [2], da ministri za šolstvo v članicah EU uvrščajo informacijske komunikacijske tehnologije med trinajst področij, ki so pomembna za vseživljenjsko izobraževanje,

lahko sklepamo, da je vsakdo, ki je uspešno končal vsaj sedmi razred osnovno šolskega izobraževanja, vsaj osnovno informacijsko pismen. Torej zna uporabljati:

- tipkovnico za vnos podatkov,
- brati in uporabljati navodila z zaslona,
- brati in uporabljati pisna navodila,
- napisati krajši pisni sestavek na izbrano temo.

Glede na to, da

- poznamo vsaj eno spletno stran in vsaj eno pisno gradivo, kjer je razumljivo opisano kako programirati procesor eProDas_Rob1, ki smo ga uporabili v naši napravi [8],
- je iz ciljev predmeta tehnika in tehnologija, ki je obvezen predmet v OŠ, jasno razvidno, da se učenci učijo o osnovnih postopkih sestavljanja gradnikov v celoto in o delovanju osnovnih elektronskih in strojnih elementov,
- je iz ciljev modula upravljanje programirljivih naprav, ki je obvezen modul, jasno razvidno, da se dijaki računalniške usmeritve v drugem letniku naučijo tehnik in postopkov programiranja,
- sta v raziskavi sodelovala dva dijaka, ki obiskujeta tretji in četrti letnik računalniške usmeritve,

lahko zaključimo, da obstaja za osnovnošolca in srednješolca ustrezno, razumljivo in dosegljivo znanje, da lahko sestavi in programira robotsko napravo, ki zna poiskati in zabiti žebelj.

5 Končno vrednotenje trditve

Trditev, ki smo jo raziskali v okviru raziskovalne naloge smo zapisali v obliki:

T1: *»Z dosegljivimi materiali in znanjem je možno narediti in programirati avtonomni sistem, ki najde in s telefonom zabije žebelj.«*

Za dokaz trditve smo morali dokazati pravilnost posameznih sklopov:

- dosegljivost materiala,
- dosegljivost in razumljivost znanja, ki smo ga potrebovali pri izgradnji in programiranju sistema,
- avtonomnost in
- funkcionalnost sistema, ki išče in s telefonom zabija žebelje.

V poglavju od 3.1 do 3.6 smo na primeru uporabe učil Lego NXT in eProDas-Rob1 dokazali, da je možno zgraditi robotski sistem za iskanje in zabijanje žebeljev na najmanj dva načina. V poglavju 3.9 smo za sistem realiziran z učilom Lego NXT dokazali funkcionalnost robotskega sistema. V poglavju 2 in 4 smo dokazali prisotnost in razumljivost fonda znanja za razumevanje samega procesa zabijanja žeblja kakor tudi za razumevanje procesa sestavljanja in programiranja avtonomne robotske naprave, ki zna ta proces udejaniti.

Iz prejšnjega odstavka je možen le en zaključek, s trditvijo se docela strinjamo in jo označimo kot pravilno

6 Zaključek

Izkušeni uporabniki kladiva si niti ne predstavljajo koliko truda in časa so vložili v piljenje veččine zabijanja žebļa. Čeprav gre za navidez enostavno opravilo, smo s študijo ustrezne literature, kar hitro ugotovili, da je za pojasnitev tega procesa potrebno kar precej fizikalnega in matematičnega znanja.

V nalogi smo raziskovali ali je možno zgraditi robotski sistem, ki bi ga preko piljenja ustreznega algoritma »naučili« te veččine. Kljub temu, da je šlo za kompleksno raziskovalno vprašanje, smo ga po delih uspešno dokazali.

Dokazali smo obstoj in dosegljivost ustreznih materialov s pomočjo katerih je možno enostavno in hitro zgraditi avtonomni robotski sistem za zabijanje žebļev. Dokazali smo obstoj in razumevanje programerskega znanja, ki je bilo potrebno za oživitev sistema. Z empiričnimi preizkusi smo tudi dokazali 93 % uspešnost zgrajenega sistema. Tako lahko raziskovalno trditev: »Z dosegljivi materiali in znanjem je možno narediti in programirati avtonomni sistem, ki najde in s telefonom zabije žebelj«, označimo kot pravilno.

Pri preizkušanju delovanja sistema smo ugotovili konstrukcijsko napako, ki jo je možno z dodatnimi stroški in dopolnitvijo programskega algoritma enostavno in hitro odpraviti. Seveda tega nismo storili, ker smo mnenja, da bi bil finančni vložek nesorazmeren dvigu kvalitete delovanja robotskega sistema.

7 Literatura in viri

- [1] CPI: Novi in prenovljeni študijski ter izobraževalni programi, Ljubljana 2006, (online), dosegljivo na <http://www.cpi.si>
- [2] European Commission - Education & training: School education: equipping a new generation, 2008, dosegljivo na http://ec.europa.eu/education/lifelong-learning-policy/doc64_en.htm
- [3] MŠŠ: Bela knjiga o vzgoji in izobraževanju v Republiki Sloveniji, Ljubljana, junij 2011
- [4] MŠŠ: Delovna področja, Osnovnošolsko, srednješolsko in višješolsko izobraževanje, MŠŠ, 2009, (online), dosegljivo na <http://www.mss.gov.si/>
- [5] Pevec G. S., Mali D.: Izhodišča za pripravo izobraževalnih programov nižjega in srednjega poklicnega izobraževanja ter programov srednjega strokovnega izobraževanja, med ključne kvalifikacije šteje delo z informacijami in informacijsko tehnologijo, Ljubljana 2006, dosegljivo na <http://www.cpi.si>
- [6] Ponikvar, D., Kokot Klančnik, T., Nemeč, J., Stojanov, T., M. Urbanč, M.: Projektno delo: Zabijanje žebnja, UL-Fakulteta za matematiko in fiziko, 2009
- [7] Programski svet za informatizacijo šolstva: Akcijski načrt nadaljnjega preskoka informatizacije šolstva, 2009
- [8] PeF, Univerza v Ljubljani, 2006: E-učbeniki za izbrane naravoslovno-tehniške predmete (online), citirano 26. 1. 2008, dostopno na <http://www.pef.uni-lj.si/narteh/>
- [9] Zakon o osnovni šoli UL št. 81,2006 – uradno prečiščeno besedilo in UL št. 102, 2007