ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA VELENJE
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA

UPORABA LEGO NXT V REALNEM SVETU

Tematsko področje: Robotika

Avtorja:
Luka Verbuč, 4. letnik
Jernej Blažič, 4. letnik

Mentor:
Maja Gačnik, dipl. ekonomist
Peter Vrčkovnik, dipl. inž. elektrotehnike

Velenje, 2012

Raziskovalna naloga je bila opravljena na Elektro in računalniški šoli Velenje.

Mentor: Maja Gačnik, Peter Vrčkovnik

Datum predstavitve:

KLJUČNA INFORMACIJSKA DOKUMENTACIJA

ŠD ŠCV 2011/2012

KG Elektrotehnika/Robotika

AV VERBUČ, Luka; BLAŽIČ, Jernej

KZ 3330 Mozirje, SLO, Brezje 60c; 2832 Mislinja, SLO, Paka – del 89

ZA Šolski center Velenje, Elektro in računalniška šola

LI 2012

IN Uporaba lego NXT v realnem svetu

TD RAZISKOVALNA NALOGA

OP

IJ SL

JI sl/en

AI

Robotika se je v zadnjih desetletjih utrdila v številnih industrijskih procesih kot nepogrešljiv del moderne, ekonomične in človeku prijazne tehnologije. Brez robotskih manipulatorjev si ne znamo več predstavljati varjenja, sestavljanja, skladiščenja …
V raziskovalni nalogi želiva v pogon spraviva na videz dve popolnoma različni robotski roki. Prvo robotsko roko bova izdelala sama, in sicer jo bova sestavila iz kompleta Lego Mindstorms. Drugo robotsko roko pa sva dobila v šoli in gre za profesionalno robotsko roko, ki se uporablja v industriji.
Za obe robotski roki bova pripravila ročno in avtomatsko krmilje. Oba programa bova primerjala in poizkušala ugotoviti, ali se da s pomočjo »igrače« Lego Mindstorms naučiti programirati tudi pravo robotsko roko.
Kot preizkus bova obe roke premikala od točke, ki jo bova določila na drugo točko. Na koncu pa bova naredila analizo, ter preverila če se lahko roka, ki je sestavljena iz kock, primerja s profesionalno roko. Programska jezika za programiranje rok sta si oblikovno zelo podobna, na podlagi tega bova ugotovila, v katerem jeziku je lažje narediti program za določeno robotsko roko in ali je en program nadgradnja drugega. Za programiranje realne robotske roke bova uporabila programski jezik BASCOM AVR, za programiranje lego robotske roke pa programski jezik C++.
Iz te raziskovalne naloge pa se bova tudi veliko naučila, saj se robotika uporablja skoraj na vsakem koraku, in to znanje, ki ga bova pridobila, nama bo prišlo še kako prav.

KEY WORDS DOCUMENTATION

ND ŠCV 2011/2012

CX Electrotechnik/Robotics

AU VERBUČ, Luka; BLAŽIČ, Jernej

PP 3330 Mozirje, SLO, Brezje 60c; 2832 Mislinja, SLO, Paka – del 89

PB Šolski center Velenje, School of Electrical Engineering and Computing

PY 2012

TI Using NXT position in the real world

DT RESEARCH WORK

NO

LA SL

AL sl/en

AB

Robotic has become a significant and indispensable part of modern, economical and user friendly technologies in the last decide. Without robot manipulators we can no longer imagine welding, assembling, storing...
In our search project we want to manipulate two robot arms that seen to be very different. The first one will be home made, from a Lego Mindstorms kit. The second one is a professional's robotic arm, very similar to those used in industry.
We will assemble a manual and automatic control for both arms. We will compare both programs try to find out, whether it is possible or not to learn how to program a real robotic arm using a Lego "toy".
As a test, we will move both arms for one point to another. At the end we will make an analysis in order to find out, if an arm mode of bricks can compare with a real arm.
Programming languages for both arms are very similar in design.. We will find out which language is easier to operate with and if one of the languages up rages another. For programming the real robotic arm we will use BASCOM-AVR and for the Lego arm, we will use C++.
We will also learn a lot from this research project, because robots are being used all around us and we are likely to benefit a lot form the knowledge.

KAZALO

KAZALO SLIK

1. Uvod

Vsa štiri leta smo se v šoli ukvarjali s programiranjem in krmiljenjem električnih naprav. Zato sva za raziskovalno nalogo izbrala robotski roki. Sprva sva mislila narediti robotsko roko iz kock *Lego Mindstorms*. Kasneje pa sva se odločila, da bova v pogon spravila še pravo robotsko roko.

Najin osrednji cilj je, da v pogon spraviva dve popolnoma različni robotski roki. Prvo robotsko roko bova izdelala sama, in sicer jo bova sestavila iz kompleta *Lego Mindstorms*. Na robotsko roko bova namestila senzorje, ki bodo merili (razdaljo, dotik, glasnost, barvo) in s tem bo robot natančnejši.

Drugo robotsko roko sva dobila v šoli; to je profesionalna robotska roka, ki se uporablja v podjetjih. Za to robotsko roko bova morala izvesti krmiljenje na ročni način in preko krmilnika. Ko bova roki spravila v pogon, bova napisala program za vsako roko. Za obe robotski roki bova naredila več posameznih programov, ki jih bova na koncu primerjala; naredila bova tudi analizo ter preverila, če se lahko roka, ki je sestavljena iz kock, primerja s profesionalni roko. Najbolj pa naju bo zanimalo ali lahko nekdo, ki napiše program za Lego robotsko roko, z malo dodatnega znanja napiše tudi program za industrijsko robotsko roko.

Ob raziskovalni nalogi se bova tudi veliko naučila in to znanje bova lahko uporabila tudi v prihodnosti, saj se robotske roke vse več uporabljajo v podjetjih.

Hipoteze

1. Če programiramo lego robotsko roko s programskim jezikom C++, s programiranjem realne robotske roke, ne bo težav.
2. Obe robotski roki imata enako zgradbo in s tem podobno delovanje.
3. Delovanje senzorjev in izvršnih členov je na obeh robotskih rokah enako.
4. Otroci, ki se ``igrajo´´ z *Lego NXT* in s tem tudi programirajo to inteligentno igračo, s programiranjem prave robotske roke ne bodo imeli težav.

2. Opis robotske roke NXT

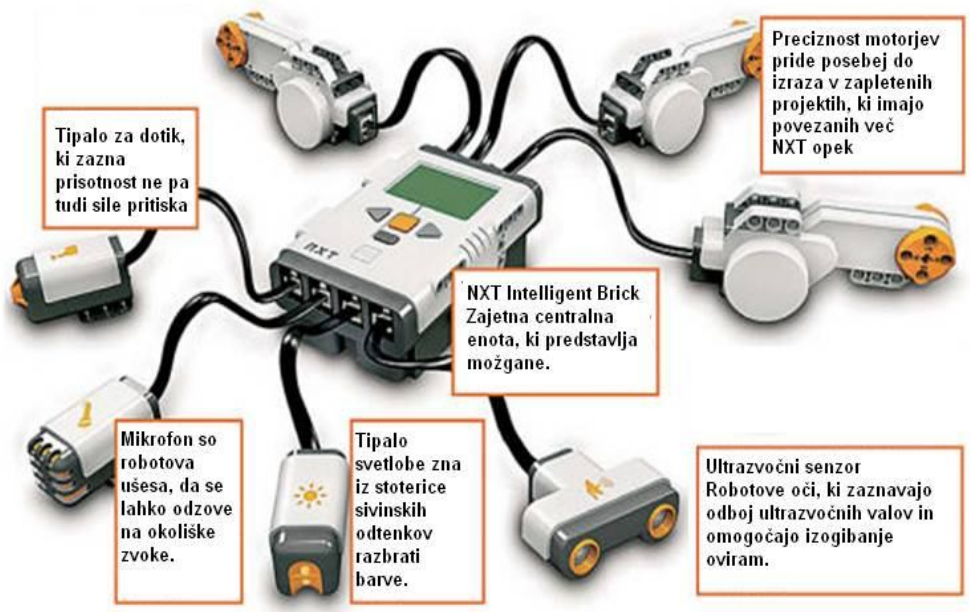
Robot *Mindstorm* se uporablja z namenom, da se otroci in starejši skozi zabavo naučijo osnov programiranja in tudi programskega jezika. Robotu dajemo ukaze preko programskega paketa. *Mindstorms NXT* programiramo v novejšem jeziku, ki je nastal leta 2006 in je zasnovan na osnovi *Lab VIEW-a*. Imenuje se *NXT-G*. Programsko okolje je grafično, kar pomeni, da se program »piše« z vlečenjem programskih gradnikov iz palete v glavno okno. Tam jih postavimo na virtualno *LEGO* letev z luknjami, ki ponazarja potek programa. Midva pa bova robota programirala še v dveh drugih programskih jezikih.

Robot *Mindstorm* je namenjen večinoma za igro in učenje programskega jezika. Drugače pa je koristen, saj se večina ljudi začne učiti s takšnimi roboti in kasneje preidejo na večje robote. Vsako leto izide projekt, s katerim je potrebno premagovati ovire in sestaviti poligon. Najina naloga, ki sva si jo zastavila, je, da sestavila robotsko roko. Ta roka naj bi bila podobna resničnemu robotu.

Slika 1: Lego robotska roka NXT

1.1 Sestavni deli lego robotske roke

Lego robotska roka je sestavljena iz lego kompleta, ki sva ga dobila v šoli. Za tega robota sva uporabila 3 servo motorje, NXT mikro-računalnik, na katerega s pomočjo priloženega USB kabla prenašaš programe iz osebnega računalnika. Za povezavo sva uporabila šest-polne ploščate kable. Ti so različnih dolžin in popolnoma izmenljivi - isti kabel je uporaben tako za priklop motorjev kakor tudi poljubnega senzorja.



Slika 2: Povezava z mikro-računalikom

3. Opis realne robotske roke

V šoli sva dobila pravo robotsko roko, ki se uporablja v podjetjih. Sestavni deli robotske roke so podnožje, ki je vrtljivo okoli navpične osi; štirje členi roke, rotacijski sklepi in motorček, s katerim pokažemo oziroma se dotaknemo želene točke. Za prve tri premike so uporabljeni motorji z reduktorjem, za 4. gib pa je uporabljen servomotor. Kot indikacija za pozicijo so uporabljeni linearni potenciometri.

Slika 3: Robotska roka

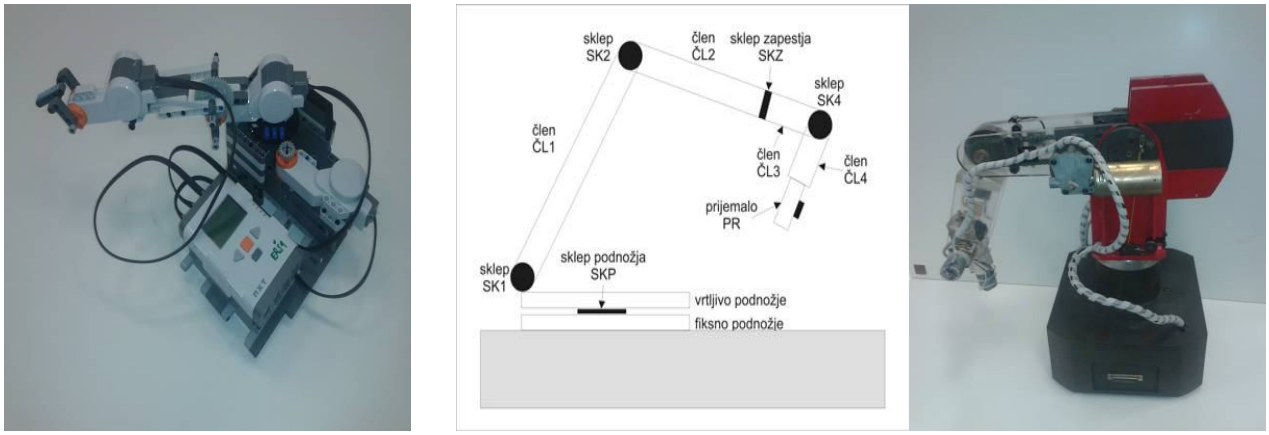
3.1 Sestavni deli realne robotske roke

Podnožje robotske roke je izdelano iz debelega železa, saj mora biti roka pri miru. Za večina členov je uporabljeno pleksi steklo. Drugi in tretji člen sta povezana do sklepa z verigo. Na robotu so vgrajena 4 končna stikala in 3 potenciometri. Končna stikala nam preprečujejo okvare na robotski roki roke. Ko pride sklep do končne lege se aktivira stikalo in prekine tokokrog. Z pomočjo potenciometrov lahko določimo v katerem položaju se nahaja določen sklep na robotski roki. Vse skupaj je povezano s kabli do podnožja, od tam pa s podaljškom povežemo do krmilne plošče.

Slika 4: Robotska roka

4. Osi robotskih rok

Vendar pa ni bilo tako enostavno, kot vsa predvidevala, saj te kocke niso tako močne, da bi roko v iztegnjeni poziciji držalo, da bi se roka lahko vrtela. Pri zobniku, ki je vrtel roko, je prišlo do takšnega razmika, da se roka ni vrtela okoli svoje osi. Najina robotska roka, ki sva jo naredila, ima 3 osi X, Y in Z; ker ta roka ni vrhunska, ima nekatere osi zelo slabo izrabljene, saj se lahko v osi X premakne 60 mm, v osi Y se premakne za ¾ okoli svoje osi, v osi Z pa se lahko premakne za 250 mm. Ta robotska roka zgleda kot pravi robot, vendar nima toliko osi, kot jih imajo roboti v proizvodnjah.



Slika 5: Robotska roka NXT, osi robota in realna robotska roka

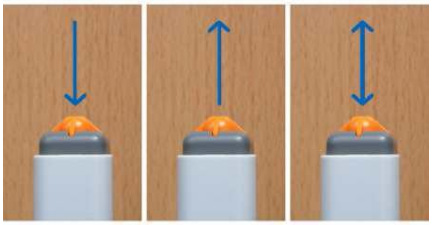
5. Senzorji

Senzorji so naprave, ki zaznavajo spremembe fizičnih količin, ki jih pretvarjajo v spremembe električnih veličin. Uporabila pa sva samo senzor dotika, da sva lahko ročno vodila lego robotsko roko. Za prepoznavanje položaja določenega sklepa ima Lego motor vgrajen inkrementalni dajalnik, kateri posreduje krmilniku podatke. V realni robotski roki sva za prepoznavanje položaja roke uporabila tri potenciometre

5.1 Senzor dotika

LEGO Mindstorms senzor dotika je preprosto stikalo, ki v obliki spremembe električnega toka zazna eno izmed treh stanj in jih posreduje inteligentni opeki. Deluje v treh položajih:

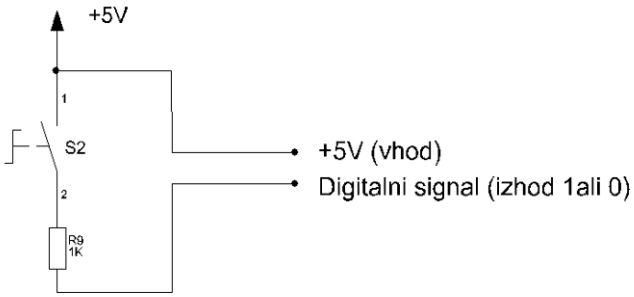
- ko se dotakne,
- ko se odmakne in
- ko je zadrževan.



Slika 6: Prikaz delovanja senzorja na dotik

Slika 7: Senzor dotika

Deluje tako, če je tipka sklenjena pošlje digitalni signal na krmilnik.



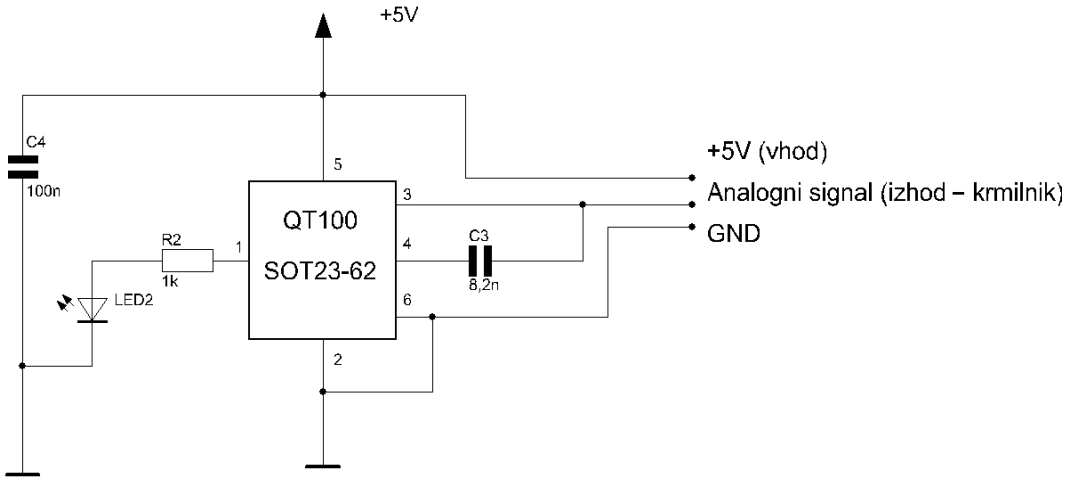
Slika 8: Shema senzorja na dotik

5.2 Svetlobni senzor

Ta senzor omogoča robotu, da meri jakost okoliške svetlobe, ki pade na njegovo fotocelico ali pa izmeri količino svetlobe, ki se je odbila od objekta, osvetljenega z LED diodo, vgrajeno v senzor.

Slika 9: Svetlobni senzor

Deluje tako, da se s spreminjanjem svetlobe spreminja tudi upornost, in prav tako tudi električni tok. LED dioda osvetljuje objekt in služi kot oddajnik. Na shemi je kondenzator (C2) zaradi enosmerne napetosti, kondenzator (C1) pa zaradi varovanja izmenične napetosti. Ima tudi dve napajanji (2 in 5). Na izhod pa gre enosmerni signal.



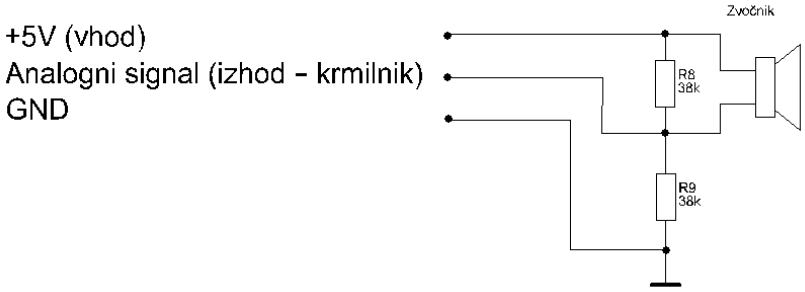
Slika 10: Shema svetlobnega senzorja

5.3 Zvočni senzor

Zvočni senzor omogoča robotu, da zaznava glasnost zvoka iz okolice, lahko pa tudi razpoznava določene tone.

Slika 11: Zvočni senzor

Deluje v frekvenčnem razponu človeškega ušesa (od 20 Hz do 20kHz), ter ta signal pošlje na krmilnik.



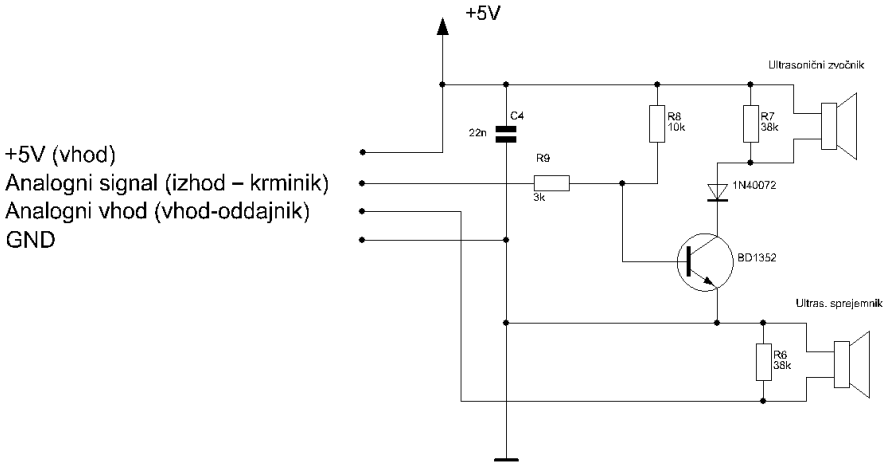
Slika 12: Shema zvočnega senzorja

5.4 Ultrasonični senzor

S pomočjo ultrasoničnega senzorja robot dokaj natančno zazna svojo oddaljenost od predmetov pred njim. Senzor je sestavljen iz dveh delov: iz oddajnika in sprejemnika.
Oddajnik oddaja zvok visoke frekvence (previsoke, da bi jo slišali), sprejemnik pa zazna, kdaj zvok v obliki valovanja pripotuje nazaj do senzorja. Na podlagi časovne razlike med oddanim in sprejetim valom lahko robot nato izračuna svojo oddaljenost od objekta pred njim. Takšno napravo ima tudi policija za merjenje hitrosti.

Slika 13: Ultrasonični senzor

Na shemi so razni elementi, ki ima vsak svojo vlogo. Dioda je, da ne pride napačna polariteta na napačno stran porabnika, ker bi lahko kateremu elementu to povzročilo škodo. Upori so za prilagajanje PIC – u, da nastavijo napetostno razliko. Elektrolitski kondenzator je namenjen temu, da vezje ima vedno dovolj napetosti. Tranzistor je namenjen temu da ojača signale nad 20 kHz.

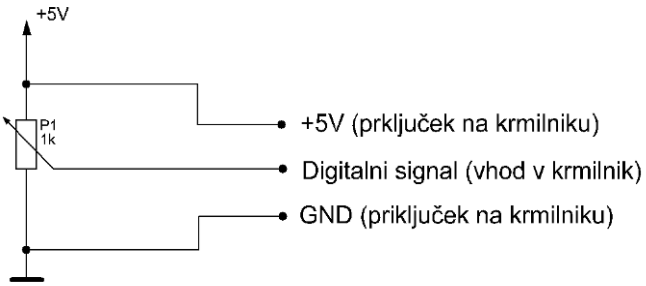


Slika 14: Shema ultrasoničnega senzorja

4.5 Senzor za realno robotsko roko

V realni robotski roki sva za senzorje uporabila tri 50kΩ potenciometre. Z premikanjem sklepa se spreminja tudi vrednost potenciometra. Potenciometre sva priključila na digitalne vhode na krmilniku. Kasneje sva v predprogramu računsko spremenila dobljene bite in z premikanjem sklepa se nama na LCD-ju prikazuje lega sklepov v stopinjah.

Slika 15: Potenciometer



Slika 16: Shema potenciometra

6. Krmilje realne robotske roke

Za krmiljenje bova uporabila dva načina:

- ročni način (z tipkami, ki so na krmilni plošči)
- avtomatski način s krmilnikom (AVR krmilnik).

5.1 Ročni način

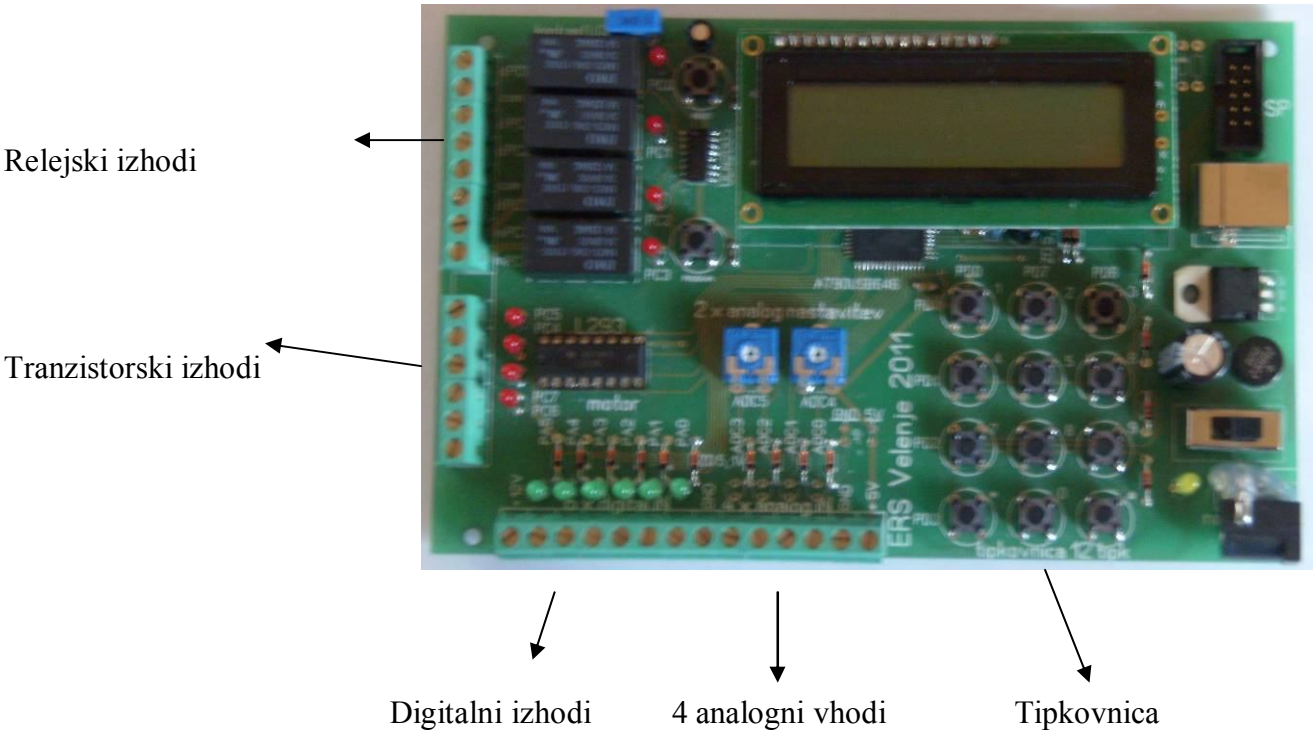
Z ročnim načinom bova krmilila robotsko roko preko tipk, ki so na krmilni plošči. Če bo člen prišel do končne pozicije, kar bo zaznalo končno stikalo ali potenciometer, se bo določen motorček ustavil. Na krmilni plošči imamo 6 tipk. Vsaka tipka za sklep ima 3 pozicije levo/sredinsko/desno. Na te tipke pa sta vezana po dva delovna kontakta. Na tipko vklop/izklop je vezan mirovni kontakt. Na desni strani je kontrolna luč, ki prikazuje, če je krmiljenje vklopljeno. Te tipke so preko vodnikov povezane do sponk. Na drugo stran sponk so priključeni releji, ki preklapljajo oz. vklapljajo motorje, ki so na robotski roki.

Slika 17: Krmilna plošča

5.2 Avtomatski način z AVR krmilnikom

Krmilno vezje je sestavljeno iz krmilnika AVR, ki krmili celotno konstrukcijo in hkrati skrbi za komunikacijo med krmilnim vezjem in računalnikom, s pomočjo katerega je nadzorovan in določen položaj roke. Krmilnik sva izdelala v 3. letniku.

Na krmilniku imamo čip L293, s katerim krmilimo realno robotsko roko. Za komunikacijo med računalnikom in krmilnikom imamo USB priključek. Na desni strani krmilnika imava relejske izhode, ki dajo kontakte. Na desni spodaj pa so 4 tranzistorski izhodi. Spodaj so sponke - 4 so za vhode, 4 pa za analogne vhode. Na krmilniku je še tipkovnica in LCD prikazovalnik.



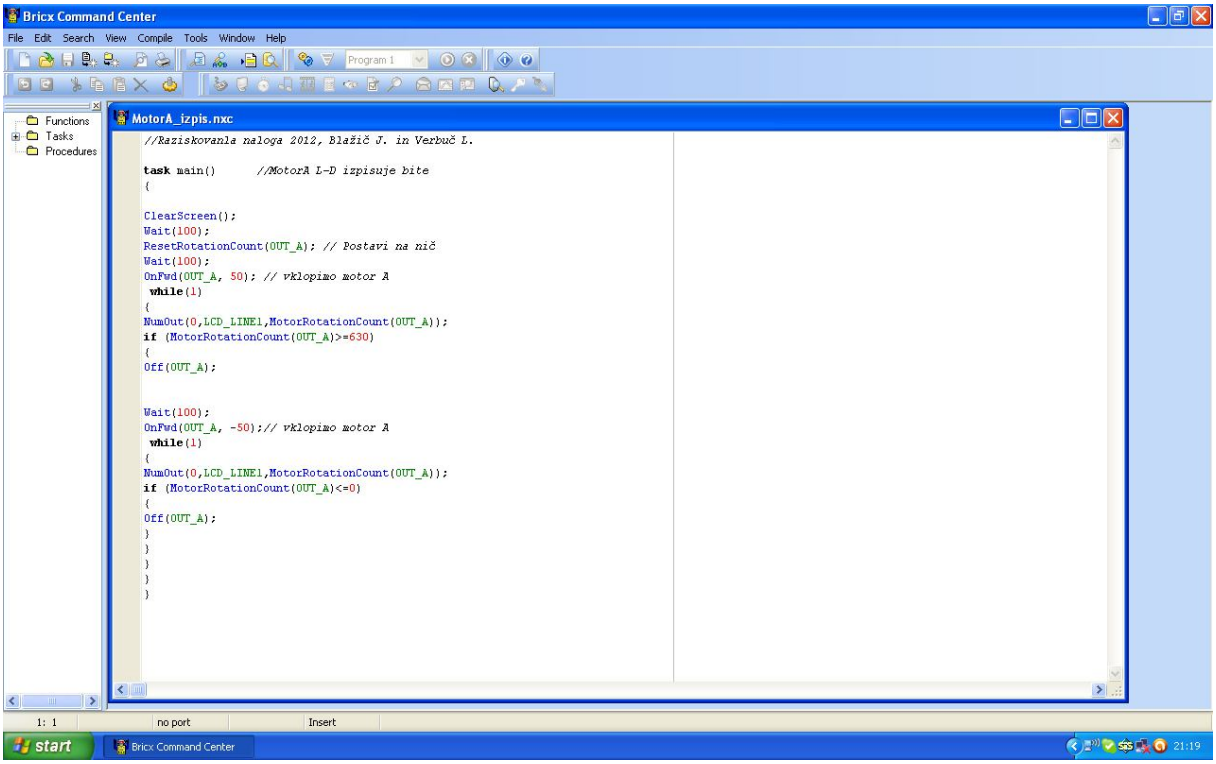
Slika 18: Krmilnik AVR

6.3 Programski jezik BASCOM

Pravo robotsko roko bova krmilila s programskim jezikom BASCOM. Najprej sva v predprogramu določila vhode ter izhode. Uporabila bova 4 relejske izhode ter 2 tranzistorska izhoda. Določila pa sva še analogne vhode, s katerimi bova vključila in izključila delovanje robotske roke. Na analogne vhode pa bova priključila še končna stikala, ki so nameščena na robotski roki in omejujejo gibanje roke.

6.4 Programski jezik C++

Programski jezik C++ je splošno namenski programski jezik, za programiranje Lego NXT. Najprej sva uporabljala originalen programski jezik, ki je bil plačljiv in ga nisva uporabljala za domače delo.



Slika 19: Programski jezik C++

7. Preizkus programa

Najprej sva spoznala programski paket *RobotC (C++),* s katerim sva pisala program za vodenje robota. Nato sva napisala osnovni program in ga naložila na robota preko USB priključka. Sledilo je pisanje programa, s pomočjo katerega se je robot peljal naprej. To sva dosegla tako, da sta se motorčka, ki sta bila priključena na B in C priključek na robotu, vrtela hkrati, in to 2 sekundi. Potem sva še napisala program, pri katerem sta se motorčka hkrati peljala naprej 2 sekundi, nato pa sta se v obratno smer peljala še nazaj, prav tako 2 sekundi.

Ker pa je ta programski jezik, ki sva ga uporabljala v šoli, plačljiv, sva morala na spletu poiskati podoben, a brezplačen program. Zaradi tega je pri ponovnem programiranju prišlo do manjših razlik.

Tukaj sva napisala program pri katerem se je motor peljal v levo do polovice, nato je počakal 2 sekundi in motor se je zapeljal v desno do polovice. Programi so v različnih programskih jezikih.

```
//Raziskovalna naloga 2012, Blažič J. in Verbuč L. (1 programski jezik C++)
task main()
 {
 motor[motorA]=50;                    //Motor se zavrti s 50% močjo
 wait1Msec(2000);                     //Vrti se 2 sekundi
 motor[motorA]=-50;                   //Motor se zavrti v drugo smer s 50% močjo
 wait1Msec(2000);                     //Vrti se 2 sekundi
 }
//Raziskovalna naloga 2012, Blažič J. in Verbuč L. (2 programski jezik C++, ki sva ga uporabljala v
nadaljevanju)
task main()                                          //Vklop motor A levo in desno
{
RotateMotor(OUT_A, 50, 640);                         //Vklop motor A desno s 50% hitrostjo
Wait(2000);                                          //Počaka 1 sekundo

RotateMotor(OUT_A, -50, 640);                        //Vklop motor A levo 50% hitrostjo
Wait(2000);                                          //Počaka 1 sekundo
}
//Raziskovalna naloga 2012, Blažič J. in Verbuč L. (programski jezik Bascom)

Do                                                   //Program naj se izvaja
Set Alevo                                            //Motor A se zavrti v levo
Waitms 2000                                          //Motor A se 2 sekundi
End if
Set Adesno                                           //Motor A se zavrti v desno
Waitms 2000                                          //Motor A se 2 sekundi
End if
End
Loop                                                 //Program naj se ponavlja
```

8. Program za lego robotsko roko

8.1 Program levo/desno z izpisom

Ta deluje tako: ko pritisneš tipko *Run* na NXT kocki sprožimo program, da se motor A najprej zavrti v levo za 640°, kar pa je v osi robota 180°, počaka 0,1 sekundo nato se zavrti v desno za 640°. Pri tem pa se izpisujejo stopinje za koliko se je moral zavrteti motor, vendar se je moral motor zavrteti za več stopinj, ker je na motorju manjši zobnik, kot pa na samem sklepu.

```
//Raziskovalna naloga 2012, Blažič J. in Verbuč L.

task main()                                          //Motor A  izpisuje stopinje
{
ClearScreen();
Wait(100);                                           //Počaka o,1 sekundo
ResetRotationCount(OUT_A);                           //Postavi na nič
Wait(100);                                           //Počaka o,1 sekundo
OnFwd(OUT_A, 50);                                    //Vklopimo motor A
 while(1)
{
NumOut(0,LCD_LINE1,MotorRotationCount(OUT_A));       //Izpis bitov v 1 vrstici
if (MotorRotationCount(OUT_A)>=640)                  //Motor A vrti do 640 stopinj ter izklopi
{
Off(OUT_A);                                          //Motor A izklopi

Wait(1000);                                          //Počaka 1 sekundo
OnFwd(OUT_A, -50);                                   //Vklopimo motor A
 while(1)
{
NumOut(0,LCD_LINE1,MotorRotationCount(OUT_A));       //Izpis bitov v 1 vrstici
if (MotorRotationCount(OUT_A)<=0)                    //Motor A vrti do 0 stopinj ter izklopi
{
Off(OUT_A);                                          //Motor A izklopi
}
}
}
}
}
```

8.2 Program za cel postopek z izpisom

Ta program deluje tako, ko pritisneš tipko *Run* se izvede naslednji cikel. Najprej se zavrti motor B za 600° navzdol, nato se motor C zavrti za 90° navzdol in nato nazaj v prvotno stanje. Po tem se motor B zavrti nazaj v prvotno stanje in se motor A zavrti za 640°, in se ponovi vse od začetka in motor A preide v prvotno stanje.

```
//Raziskovalna naloga 2012, Blažič J. in Verbuč L.

task main()                                          //Motor ABC  izpisuje stopinje
{
ClearScreen();
Wait(100);                                           //Počaka o,1 sekundo
ResetRotationCount(OUT_B);                           //Postavi na nič
Wait(100);                                           //Počaka o,1 sekundo
OnFwd(OUT_B,-30);                                    //Vklopimo motor B
 while(1)
{
NumOut(0,LCD_LINE2,MotorRotationCount(OUT_B));       //Izpis bitov v 2 vrstici
if (MotorRotationCount(OUT_B)<=-550)                 //Motor B vrti do -600 stopinj
{
Off(OUT_B);                                          //Motor B izklopi

ClearScreen();
Wait(100);                                           //Počaka o,1 sekundo
ResetRotationCount(OUT_C);                           //Postavi na nič
Wait(100);                                           //Počaka o,1 sekundo
OnFwd(OUT_C, 10);                                    //Vklopimo motor C
 while(1)
{
NumOut(0,LCD_LINE3,MotorRotationCount(OUT_C));       //Izpis bitov v 3 vrstici
if (MotorRotationCount(OUT_C)>=75)                   //Motor C vrti do 90 stopinj
{
Off(OUT_C);                                          //Motor C izklopi

Wait(100);                                           //Počaka o,1 sekundo
OnFwd(OUT_C, -10);                                   //Vklopimo motor C
 while(1)
{
NumOut(0,LCD_LINE3,MotorRotationCount(OUT_C));       //Izpis bitov v 3 vrstici
if (MotorRotationCount(OUT_C)<=0)                    //Motor C vrti do 0 stopinj
{
Off(OUT_C);                                          //Motor C izklopi

Wait(100);                                           //Počaka o,1 sekundo
OnFwd(OUT_B, 30);                                    //Vklopimo motor B
 while(1)
{
NumOut(0,LCD_LINE2,MotorRotationCount(OUT_B));       //Izpis bitov v 2 vrstici
if (MotorRotationCount(OUT_B)>=0)                    //Motor B vrti do 0 stopinj
{
```

```
Off(OUT_B);                                   //Motor B izklopi
ClearScreen();
Wait(100);
ResetRotationCount(OUT_A);                    //Postavi na nič
Wait(100);
OnFwd(OUT_A, 50);                             //Vklopimo motor A
 while(1)
{
NumOut(0,LCD_LINE1,MotorRotationCount(OUT_A));  //Izpis bitov v 1 vrstici
if (MotorRotationCount(OUT_A)>=640)           //Motor A vrti do 630 stopinj
{
Off(OUT_A);                                   //Motor A izklopi

ClearScreen();
Wait(100);                                    //Počaka o,1 sekundo
ResetRotationCount(OUT_B);                    //Postavi na nič
Wait(100);                                    //Počaka o,1 sekundo
OnFwd(OUT_B,-30);                             //Vklopimo motor B
 while(1)
{
NumOut(0,LCD_LINE2,MotorRotationCount(OUT_B));  //Izpis bitov v 2 vrstici
if (MotorRotationCount(OUT_B)<=-550)          //Motor B vrti do -600 stopinj
{
Off(OUT_B);                                   //Motor B izklopi

ClearScreen();
Wait(100);                                    //Počaka o,1 sekundo
ResetRotationCount(OUT_C);                    //Postavi na nič
Wait(100);                                    //Počaka o,1 sekundo
OnFwd(OUT_C, 10);                             //Vklopimo motor C
 while(1)
{
NumOut(0,LCD_LINE3,MotorRotationCount(OUT_C));  //Izpis bitov v 3 vrstici
if (MotorRotationCount(OUT_C)>=75)            //Motor C vrti do 90 stopinj
{
Off(OUT_C);                                   //Motor C izklopi

Wait(100);                                    //Počaka o,1 sekundo
OnFwd(OUT_C, -10);                            //Vklopimo motor C
 while(1)
{
NumOut(0,LCD_LINE3,MotorRotationCount(OUT_C));  // Izpis bitov v 3 vrstici
if (MotorRotationCount(OUT_C)<=0)             //Motor C vrti do 0 stopinj
{
Off(OUT_C);                                   //Motor C izklopi

Wait(100);                                    //Počaka o,1 sekundo
OnFwd(OUT_B, 30);                             //Vklopimo motor B
 while(1)
{
NumOut(0,LCD_LINE2,MotorRotationCount(OUT_B));  //Izpis bitov v 2 vrstici
if (MotorRotationCount(OUT_B)>=0)             //Motor B vrti do 0 stopinj
{
```

```
Off(OUT_B);                                          //Motor B izklopi

Wait(100);
OnFwd(OUT_A, -50);                                   //Vklopimo motor A
 while(1)
{
NumOut(0,LCD_LINE1,MotorRotationCount(OUT_A));       //Izpis bitov v 1 vrstici
if (MotorRotationCount(OUT_A)<=0)                    //Motor A vrti do 0 stopinj
{
Off(OUT_A);                                          //Motor A izklopi

}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
}
```

9. Program v Bascomu

V pred programu sva določila vhodne in izhodne veličine.

```
//Raziskovalna naloga 2012, Verbuč L. in Blažič J

Alevo       Alias Portc.0                  //Določila sva, relejske izhode
Adesno      Alias Portc.1
Bgor        Alias Portc.2
Bdol        Alias Portc.3

Cdol        Alias Portc.4                  //Določila sva, tranzistorske izhode
Cgor        Alias Portc.5
Vhod1       Alias Porta.0                  //Določila sva Vhod1
Vhod2       Alias Porta.1                  //Določila sva Vhod1
Vhod3       Alias Porta.2                  //Določila sva Vhod1

1v = Avhod1                                //Dobljene biti so enaki spremenljivki 1v
1l = 1v / 3.1                              //180 stopinj dobimo če od spremenljivke 1v delim 3.1
Asklep = 1l – 130                          //Tako dobim vrednosti od 0-180 stopinj

2v = Avhod2                                //Dobljene biti so enaki spremenljivki 2v
2l = 2v / 3.1                              //180 stopinj dobimo če od spremenljivke 2v delim 3.1
Bsklep = 2l – 130                          //Tako dobim vrednosti od 0-180 stopinj

3v = Avhod3                                //Dobljene biti so enaki spremenljivki 3v
3l = 3v / 3.1                              //180 stopinj dobimo če od spremenljivke 3v delim 3.1
Csklep = 3l – 130                          //Tako dobim vrednosti od 0-180 stopinj

If Asklep < 100 And Asklep > 95 The        // Ko bo prišel Asklep med 95-100 stopinj
Cls                                        // Se bo izpis na LCD osvežil
End If

If Asklep < 13 And Asklep > 7 Then         //Ko bo prišel Asklep med 7-13 stopinj
Cls                                        // Se bo izpis na LCD osvežil
End If

If Asklep < 185 And Asklep > 700000 The    //Ko bo prišel Asklep med 185-700000 stopinj
Cls                                        // Se bo izpis na LCD osvežil
End If

If Bsklep < 100 And Bsklep > 95 The        /Ko bo prišel Asklep med 95-100 stopinj
Cls                                       // Se bo izpis na LCD osvežil
End If

If Bsklep < 13 And Bsklep > 7 Then         //Ko bo prišel Asklep med 7-13 stopinj
Cls                                        // Se bo izpis na LCD osvežil
End If

If Bsklep < 185 And Bsklep > 700000 The    //Ko bo prišel Asklep med 185-700000 stopinj
Cls                                        // Se bo izpis na LCD osvežil
End If

If Csklep < 100 And Csklep > 95 Then       //Ko bo prišel Asklep med 95-100 stopinj
Cls                                              // Se bo izpis na LCD osvežil
```

```
End If

If Csklep < 13 And Csklep > 7 Then          //Ko bo prišel Asklep med 7-13 stopinj
Cls                                         // Se bo izpis na LCD osvežil
End If

If Csklep < 185 And Csklep > 700000 Then    //Ko bo prišel Asklep med 185-700000 stopinj
Cls                                         // Se bo izpis na LCD osvežil
End If

Locate 1 ,  1                               //Dobljene stopinje se nam izpisujejo na LCD
Lcd Csklep

Locate 1 , 5                                //Dobljene stopinje se nam izpisujejo na LCD
Lcd Csklep

Locate 2 ,  1                               //Dobljene stopinje se nam izpisujejo na LCD
Lcd Csklep
End
```

9.1 Program levo/desno z izpisom

Ko sva določila vhode ter izhode, sva napisala prvi program. Program deluje tako, da se prvi motor zavrti v levo in tako se tudi prvi sklep na roki zavrti v levo in po 500 ms se motor ustavi. Nato isti korak naredi še v desno stran. Ko konča prvi, se začne premikati drugi in kasneje še 3 člen. S tem programom sva hotela preveriti, če se bodo sklepi vrteli v prave smeri.

```
//Raziskovalna naloga 2012, Verbuč L. in Blažič J

Do

Set Alevo                    //Vključi se motor A in sklep gre levo
Waitms 500                   //Motor naj deluje 500 ms
Reset  Alevo                 //Izključi se motor A

Set Adesno                   //Vključi se motor A in sklep gre desno
Waitms 500                   //Motor naj deluje 500 ms
Reset  Adesno                //Izključi se motor A

Set Bgor                     //Vključi se motor B in sklep gre gor
Waitms 500                   //Motor naj deluje 500 ms
Reset  Bgor                  //Izključi se motor B

Set Bdol                     //Vključi se motor B in sklep gre dol
Waitms 500                   //Motor naj deluje 500 ms
Reset  Bdol                  //Izključi se motor B

Set Cgor                     //Vključi se motor C in sklep gre gor
Waitms 500                   //Motor naj deluje 500 ms
Reset  Cgor                  //Izključi se motor C

Set Cdol                     //Vključi se motor C in sklep gre dol
Waitms 500                   //Motor naj deluje 500 ms
Reset  Cdol                  //Izključi se motor C

End
Loop                         //Program naj se ponavlja
```

9.2 Program za cel postopek z izpisom

Ta program deluje tako, ko pritisneš tipko 1 se izvede naslednji cikel. Najprej se zavrti motor B za 180° navzdol, nato se motor C zavrti za 90° navzdol in nato nazaj v prvotno stanje. Po tem se motor B zavrti nazaj v prvotno stanje in se motor A zavrti za 180°, in se ponovi vse od začetka in motor A preide v prvotno stanje.

```
//Raziskovalna naloga 2012, Verbuč L. in Blažič J

Gosub Tipkovnica                    //Program naj skoči v predprogram tipkovnica

If Tipka = 1 Then                   //Če bo prva tipka na tipkovnici 1
Set Bdol                            //Se bo vključil B motor in sklep bo šel dol
End If

If Bsklep > 88 And Bsklep < 92 Then          //Če bo sklep prišel med 178-180 stopinjami
2i = 1                                       //Se bo spremenljivka 2i postavila na vrednost 1
End If

If 2i = 1 Then                               // Če bo spremenljivka 2i na1
Reset Bdol                                   //Se bo motor B ustavil
Set Cdol                                     //Se bo motor C vključil in sklep bo šel dol
End if

If Csklep > 88 And Csklep < 92 Then          //Če bo sklep prišel med 88-92 stopinjami
3i = 1                                       //Se bo spremenjlivka 3i postavila na 1
End If

If 3i = 1 Then                               // Če se bo spremenljivka 3i postavila na vrednost 1
Reset Cdol                                   //Se bo motor C ustavil
2i=0                                         //Spremenljivka 2i se bo postavila na 0
Set Cgor                                     //Se bo motor C vključil in sklep bo šel gor
End If

If Csklep > 0 And Csklep < 2 Then      // Če bo sklep prišel med 88-92 stopinjami
Reset Cgor                                   //Se bo motor C ustavil
3i = 0                                 //Spremenljivka 3i se bo postavila na 0
Set Bgor                                     //Se bo motor B vključil in sklep bo šel gor
End If

If Bsklep > 0 And Bsklep < 2 Then            //Če bo sklep prišel med 0-2 stopinjami
Reset Bgor                                   //Se bo motor B ustavil
Set Adesno                          //Se bo motor A vključil in sklep bo šel desno
End If

If Asklep > 178 And Asklep < 182 The         // Če bo sklep prišel med 0-2 stopinjami
1i = 0                                       //Spremenljivka 1i se bo postavila na 1
End If

If 1i = 1 Then                               // Če se bo spremenljivka 1i postavila na vrednost 1
Reset Adesno                                 //Se bo motor A ustavil
```

```
Set Bdol                                    //Se bo vključil motor B in sklep bo šel dol
End If

If Bsklep > 88 And Bsklep < 92 Then         // Če bo sklep prišel med 88-92 stopinjami
2i = 1                                      //Spremenljivka 1i se bo postavila na 1
End If


If 2i = 1 Then                              // Če bo spremenljivka 2i na vrednosti 1
Reset Bdol                                  // Se bo ustavil motor B
Set Cdol                                    //Se bo vključil motor C in sklep bo šel dol
End if

If Csklep > 88 And Csklep < 92 Then         // Če bo sklep prišel med 88-92 stopinjami
3i = 1                                      //Spremenljivka 3i se bo postavila na 1
End If

If 3i = 1 Then                              // Če bo spremenljivka 3i na vrednosti 1
Reset Cdol                                  // Se bo ustavil motor C
2i=0                                        // Se bo spremenljivka 2i postavila na 0
Set Cgor                                    // Se bo vključil motor C in sklep bo šel gor
End If

If Csklep > 0 And Csklep < 2 Then           // Če bo sklep prišel med 0-2 stopinjami
Reset Cgor                                  // Se bo ustavil motor C
3i = 0                                      // Se bo spremenljivka 3i postavila na 0
Set Bgor                                    // Se bo vključil motor Bin sklep bo šel gor
End If

If Bsklep > 0 And Bsklep < 2 Then           // Če bo sklep prišel med 0-2 stopinjami
Reset Bgor                                  // Se bo ustavil motor b
2i = 0                                      // Se bo spremenljivka 2i postavila na 0
Set Alevo                                   // Se bo vključil motor B in sklep bo šel levo
End If

If Asklep > 0 And Asklep < 2 Then           // Če bo sklep prišel med 0-2 stopinjami
1i = 1                                      // Se bo spremenljivka 1i postavila na 1

End If

If 1i = 1 Then                              // Če bo spremenljivka 11 na vrednosti 1
Reset Alevo                                 // Se bo motor A ustavil
End If

Loop                                        //Program naj se ponavlja
End                                         //Konec
```

10. Primerjava programov – ukazov

Začetek in konec programa

- Bascom Do, Loop, End
- C++ **task** main()

V programskem jeziku C++ pričnemo z pisanjem programa z ukazom (**task** main()), kar pomeni začetek programa, v jeziku Bascom pa z ukazom(Do), kar pomeni začetek programa. Konec programa pa zaključimo z End, Loop ukazom pa ukažemo da naj se program ponavlja. Začetek in konec programa se v programskima jezikoma razlikujeta, zato potrebujemo za uporabo ukazov predznanje.

Vklop motorjev levo desno

- Bascom Set Alevo, Reset Adesno
- C++ RotateMotor(OUT_A, 50, 640);

Za pogon motorja napišemo ukaz v C++ (RotateMotor(OUT_A, 50, 640);) za v drugo smer pa napišemo(RotateMotor(OUT_A, -50, 640);), kar pomeni zavrti motor na izhodu A, 50, -50 pomeni s kolikšno močjo se naj vrti in v katero smer, 640 pa pomeni za koliko stopinj se naj zavrti motor. V Bascomu za eno smer (Set Alevo) za drugo smer pa (Reset A desno). V Bascomu pa je slabost, ker ne moremo narediti s kolikšno močjo in za koliko se naj zavrti motor. Vklop motorja levo/desno se ne razlikujeta veliko, razlika je samo, da lahko v C++ nastaviš moč motorja v C++ so bolj zahtevani ukazi kot v Bascomu.

Čas delovanja

- Bascom Wait 2000
- C++ Wait(2000);

Za čas uporabimo v C++ (Wait(2000);) in je vedno v mili sekundah. V Bascomu pa (Waitms 2000), lahko nastavimo tudi tako Wait in podane so sekunde. Čas delovanja se v programskima jezikoma ne razlikujeta veliko.

Izpis na LCD zaslon

- Bascom Lcd Asklep, Locate 1 , 1
- C++ NumOut(0,LCD_LINE1,MotorRotationCount(OUT_A));

Za izpis za LCD zaslon v C++ sva morala napisati NumOut(0,LCD_LINE1,MotorRotationCount(OUT_A), line 1 pomeni, da naj se izpis prikaže v prvi vrstici zaslona, za koliko se motor premakne v stopinjah motor na izhodu A. V Bascomu najprej napišemo Lcd Asklep, kar pomeni, da se nam izpisuje pozicija motorja A (Asklep). Locate 1 , 1 to pomeni, da se nam izpis zapisuje v prvi vrstici prvem kvadratu. Izpis na LCD zaslonu se v programskima jezikoma ne razlikujejo.

11. Zaključek

Pri raziskovalni nalogi, sva ugotovila, da se različni programski jeziki za programiranje robotskih rok ne razlikujejo veliko. Največje razlike so v ukazih, kaj se naj zgodi. Za lego robotsko roko, ki sva jo sestavila, sva poskušala narediti čim bol podobno realni robotski roki. Ugotovila sva, da se lahko igrača kot je Lego NXT primerja z napravo, ki se uporablja v industriji. Robotski roki se razlikujeta samo po konstrukciji.

1. Če bi znali programirati s programskim jezikom C++ in bi želeli programirati realno robotsko roko v programskem jeziku Bascom, ne bi bilo potrebno vložiti veliko truda v učenje , ker je Bascom lažji. V programskem jeziku C++ aktiviramo motor z ukazom (RotateMotor(OUT_A, 50, 640);), v programskem jeziku Bascom pa (Set Alevo).

2. Robotski roki sva poskusila narediti čim bol enaki in meniva, da nama je tudi uspelo. Seveda se ne moreta primerjati po velikosti vendar obe delujeta enako. Ugotovila sva da je Lego robotska roka bolj natančna, saj lahko spreminjaš moč motorja. Lego robotska roka je natančna tudi zaradi koračnih motorjev, realna robotska roka pa ima navadne motorje. Edina slabost lego robotske roke je v tem, ker ima premajhno pomikanje po oseh XYZ. Za merjenje pozicije pri realni robotski roki sva uporabila potenciometre. Lego robotska roka pa že ima v motorju vgrajen inkrementalni dajalnik, kateri posreduje podatke za zasuk roke krmilniku.

3. Na Lego robotski roki sva uporabila samo senzorje dotika, ki sva jih uporabila pri ročnem načinu krmiljenja. Drugače jih nisva potrebovala, saj nam koračni motorji s svojimi koraki samodejno izpisujejo lego sklepov. Na realni robotski roki sva uporabila tri potenciometre s katerimi sva lahko določala na lego sklepov

4. Meniva da je programski paket Bascom lažji za razumevanje kot C++. Med samimi ukazi ne vidiva bistvene razlike med programskima jezikoma, razlika med njima je samo določenih segmentih samega ukaza. Otroci ki se igrajo z Lego NXT, ne bi imeli težav s programiranjem pravih robotskih rok.

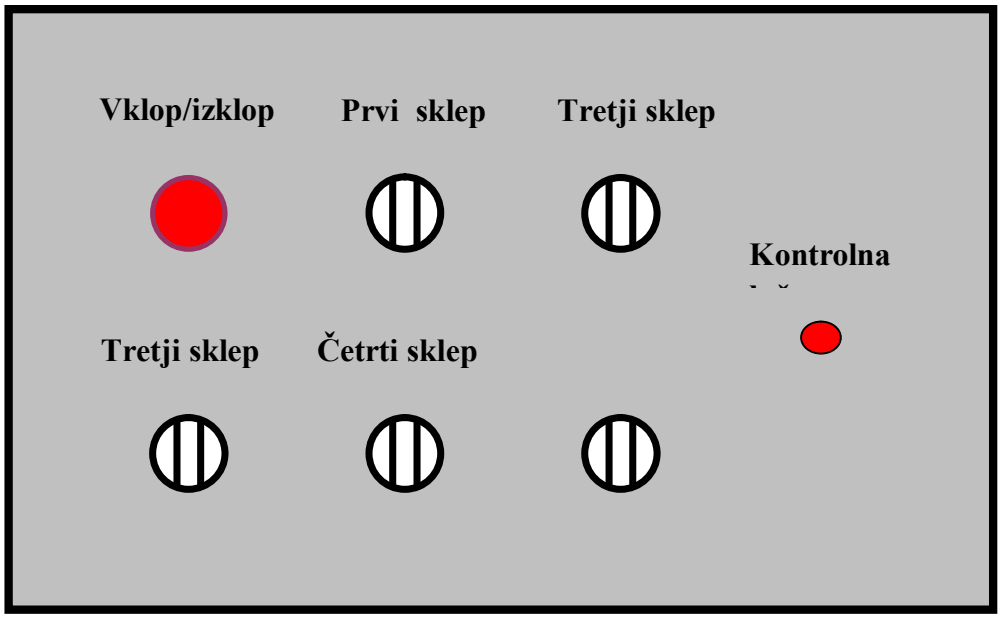
Za zaključek bi pa dodala: Otroci ki se igrajo z Lego NXT, ne bi imeli težav s programiranjem pravih robotskih rok. Seveda realne robotske roke ne bi znali priključiti in povezati z krmilno omarico, vendar bi bili sposobni napisati samo program. Če bi hoteli, da otrok napiše program v Bascomu, meniva da bi se po približno petih urah naučil samostojno napisati program za realno robotsko roko.

12. Zahvala

Zahvaljujeva se mentorju g. Petru Vrčkovniku in ga. Maji Gačnik za pomoč in usmerjanje pri izdelavi raziskovalne naloge. Zahvala gre tudi ga. Simoni Diklič za lektoriranje raziskovalne naloge, ter g. Marjanu Mlinareviću za pomoč in potrošen material.
Posebna zahvala gre tudi šoli, saj nama je priskrbela pravo robotsko roko in komplet Lego Mindstorms. Zahvala pa gre tudi staršem, ki so nama vse skozi stali ob strani.

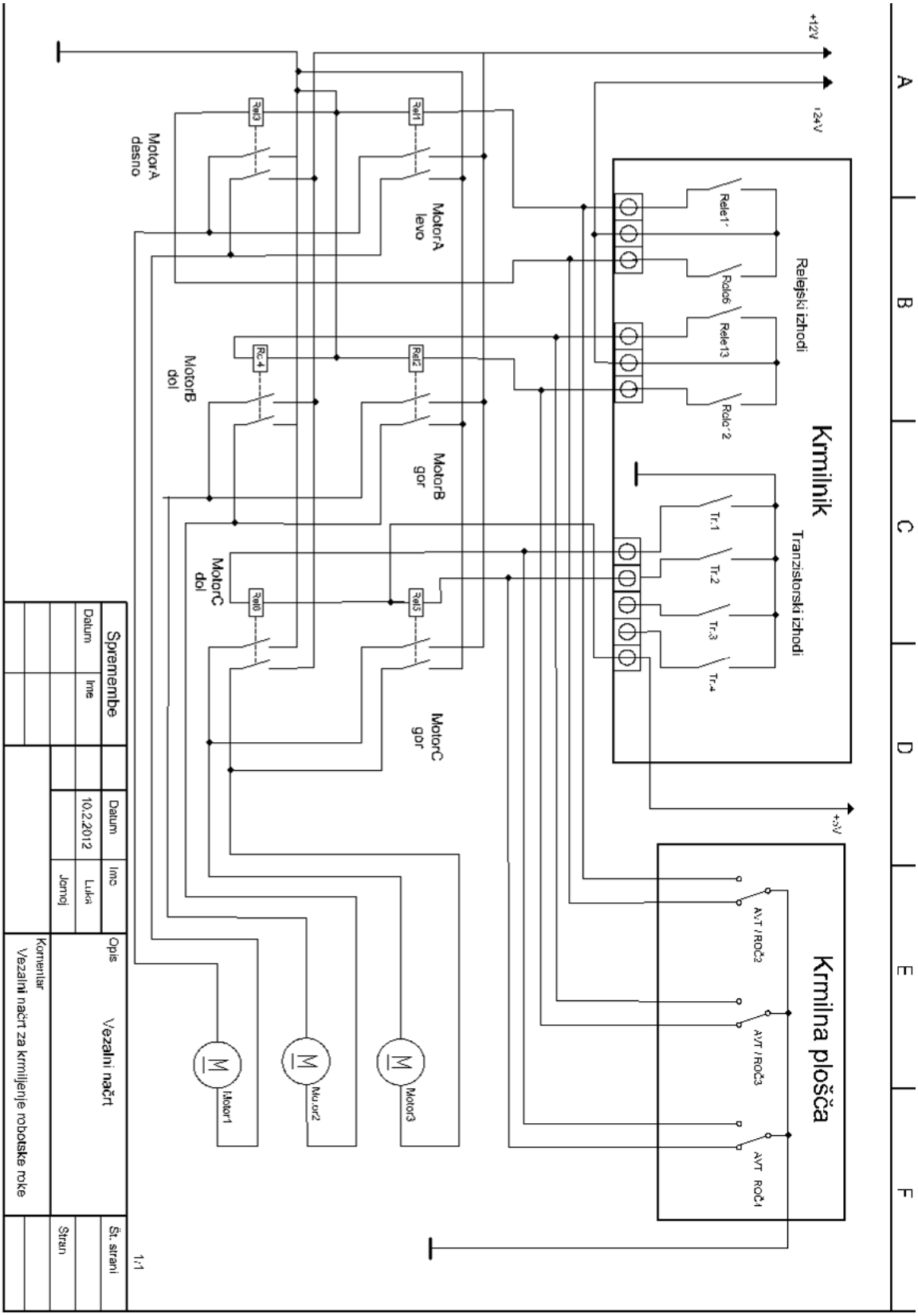
13. Priloge

13.1 Shema krmilne plošče



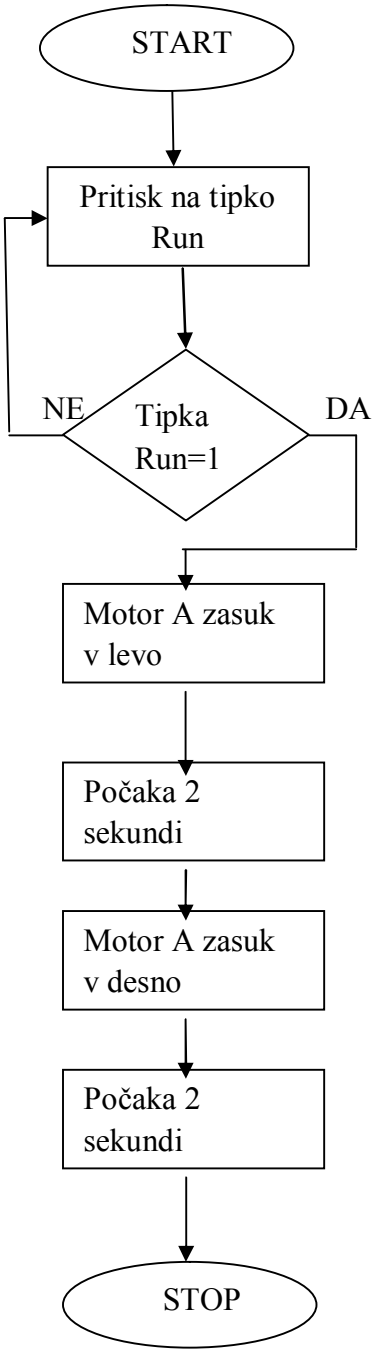
Slika 20: Shema krmilne plošče

13.2 Shema krmilne omarice za realno robotsko roko

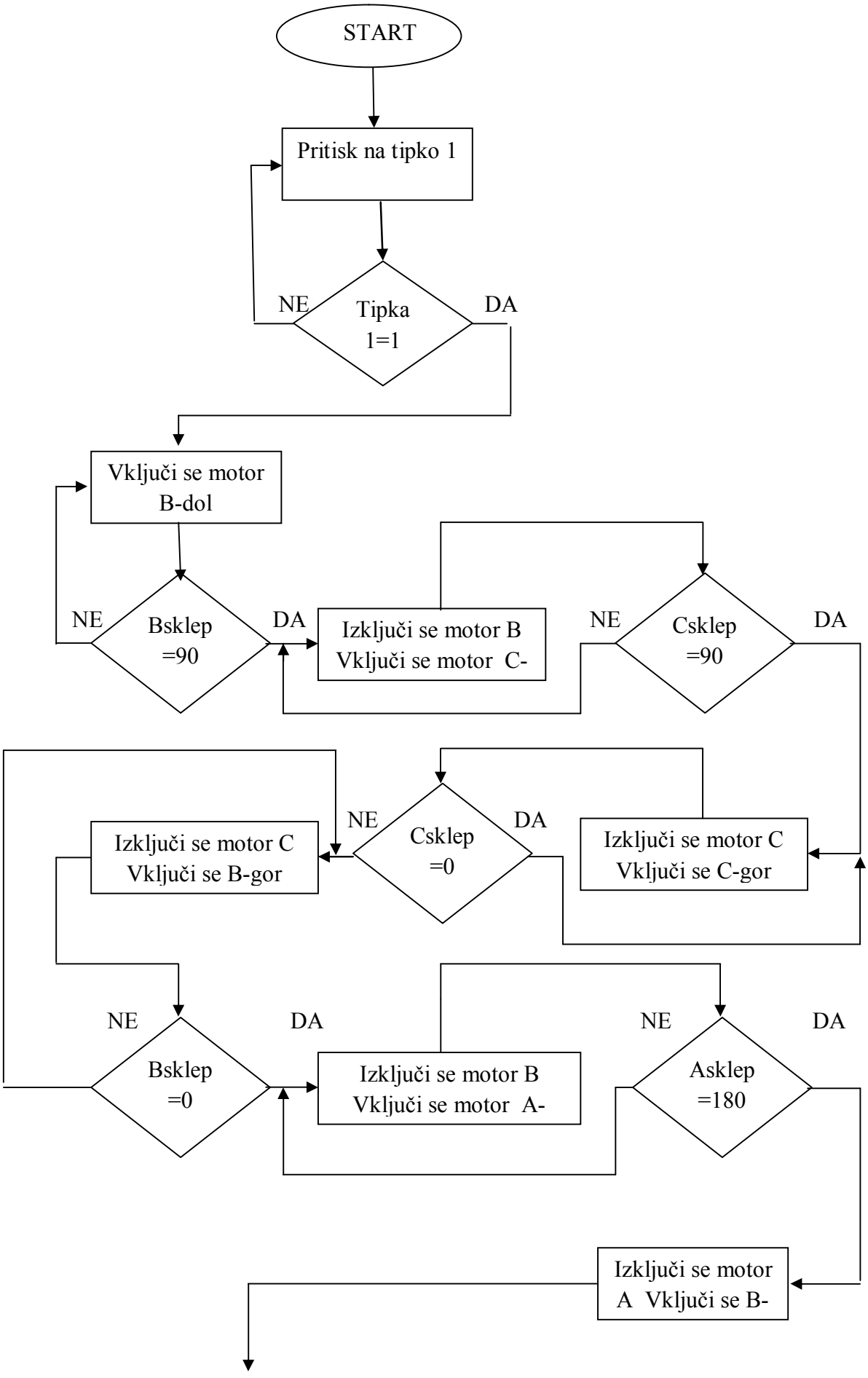


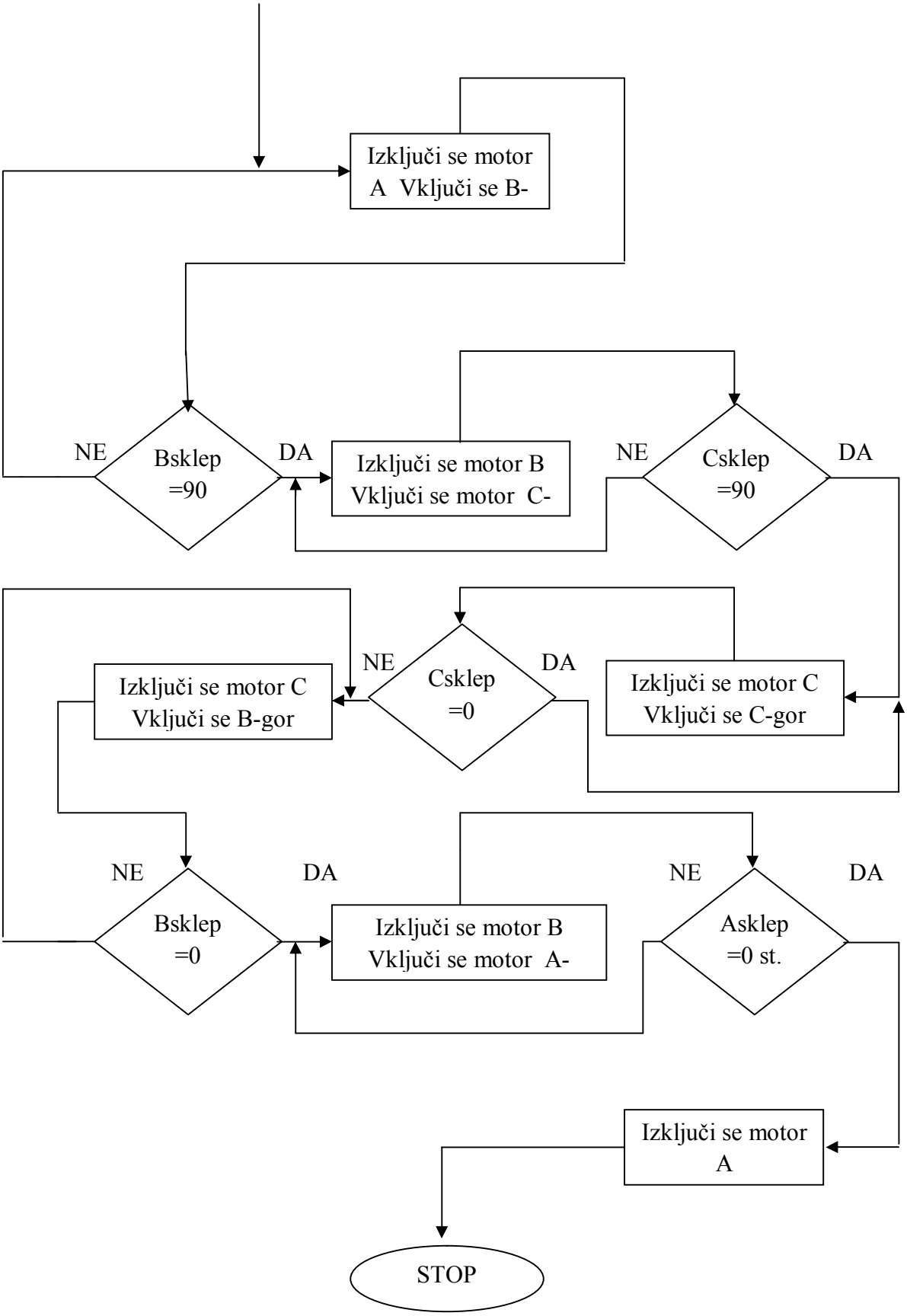
Slika 21: Vezalni načrt robotske roke

13.3 Diagram poteka za motor levo/desno



13.4 Diagram poteka za cel postopek




Izključi se motor A Vključi se B-
NE
Bsklep =90
DA
Izključi se motor B Vključi se motor C-
NE
Csklep =90
DA
NE
Csklep =0
DA
Izključi se motor C Vključi se B-gor
Izključi se motor C Vključi se C-gor
NE
Bsklep =0
DA
Izključi se motor B Vključi se motor A-
NE
Asklep =0 st.
DA
Izključi se motor A
STOP