

»Mladi za napredek Maribora 2017«

34. srečanje

Brezžično krmiljenje robota

Raziskovalno področje: **ELEKTROTEHNIKA, ELEKTRONIKA**

Raziskovalna naloga

Prostor za nalepko

Avtor: IGOR KEPE, MARTIN FERENEC
Mentor: MILAN IVIČ
Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

Maribor, januar 2017

KAZALO VSEBINE

1. POVZETEK.....	3
2. UVOD.....	3
3. VSEBINSKI DEL	4
3.1 Arduino Uno	4
3.2 Ultrazvočni senzor HC-SR04	6
3.3 Bluetooth komunikacijski modul HC-06.....	7
3.4 H-Mostič krmilnik motorjev	8
3.5 Servo pogon	9
3.6 DC Motor.....	10
3.7 Robotska roka	11
3.8 IP Kamera	12
3.9 Aplikacija.....	13
3.10 Programiranje in opis programske kode.....	14
4. ZAKLJUČEK.....	18
5. DRUŽBENA ODGOVORNOST.....	18
6. PRILOGE.....	19
6.1 Arduino programska koda	19
6.2 MIT App Inventor 2 koda.....	24
7. VIRI.....	27

KAZALO SLIK

<i>Slika 1: Arduino UNO (vir: Arduino.cc)</i>	4
<i>Slika 2: PWM Graf (vir: Arduino.cc)</i>	4
<i>Slika 3: Ultrasonični senzor HC-SR04 (vir: Google Slike)</i>	6
<i>Slika 4: Bluetooth modul HC-06 (vir: Google Slike)</i>	7
<i>Slika 5: H-Bridge motor driver (vir: Google Slike)</i>	8
<i>Slika 6: Servo pogon (vir: Google Slike)</i>	9
<i>Slika 7: DC Motor (vir: Google Slike)</i>	10
<i>Slika 8: Robotska roka (vir: avtorji naloge)</i>	11
<i>Slika 9: IP Kamera (vir: avtorji naloge)</i>	12
<i>Slika 10: Aplikacija (vir: avtorji naloge)</i>	13
<i>Slika 11: Programsko okolje MIT App Inventor 2 (vir: avtorji naloge)</i>	16

1. POVZETEK

V tej raziskovalni nalogi smo raziskovali robota. Uporabili smo odprtokodno strojno opremo Arduino. Robot ima štiri kolesa, na vsakem kolesu en motor. Motorje z Arduino kontroliramo preko krmilnika motorjev H-Mostič. S pomočjo Bluetooth modula, ki je priključen na Arduino, komuniciramo med Arduino in telefonom. Na telefonu se izvaja program, preko katerega pošiljamo posamezne ukaze Arduino. Aplikacija na telefonu je napisana v programskem okolju MIT App Inventor 2. Zraven tega pa je na robotu še IP kamera, ki preko Wi-Fi omogoča direkten prenos videa v aplikacijo na telefonu. S tem omogočamo krmiljenje robota tudi takrat, ko ni v vidnem polju, kar v praksi omogoča številne možnosti uporabe.

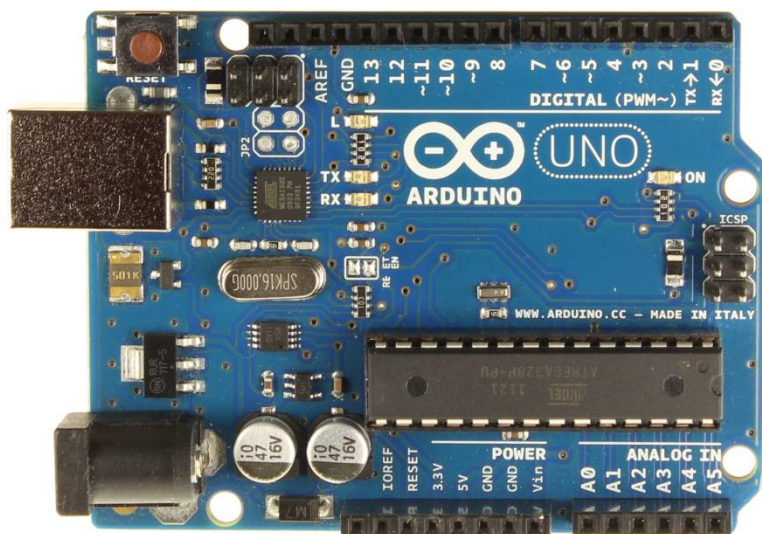
2. UVOD

Na začetku smo sestavili podvozje robota in namestili motorje, mikrokrmilnik in krmilnik motorjev H-Mostič. Tako smo preizkusili kako se krmili H-Mostič in preizkusili smo različne hitrosti vožnje. Nato smo dodali Bluetooth modul in napisali aplikacijo za telefon z operacijskim sistemom Android. Na začetku smo krmilili samo motorje, kasneje pa smo se odločili, da dodamo robotsko roko. Po preizkušnji robotske roke smo dodali še ultrazvočni senzor, ki bi preprečil morebitne poškodbe na robotu tako, da se robot ustavi, ko se približa oviri na določeni razdalji. Na koncu smo dodali še IP kamero, ki omogoča prenos videa med robotom in aplikacijo. To nam omogoča upravljanje robota tudi takrat, ko ga ne vidimo.

Komunikacija med Bluetooth modulom in aplikacijo je bila zapletena, vendar smo jo hitro razumeli.

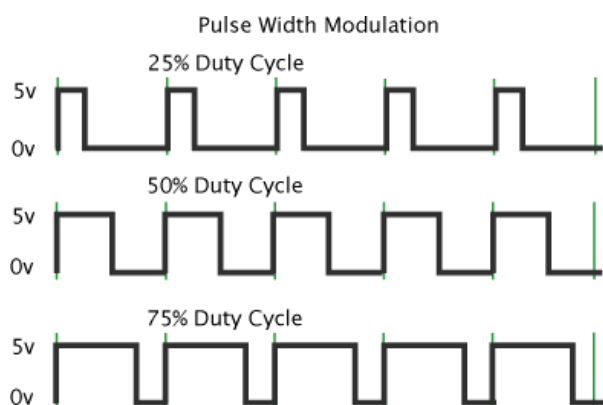
3. VSEBINSKI DEL

3.1 Arduino Uno



Slika 1: Arduino UNO (vir: Arduino.cc)

Arduino Uno je odprtokodni mikrokrmilnik, ki so ga razvili v Italiji. Ima 14 digitalnih vhodov ali izhodov (6 od teh je PWM izhodov). PWM angleško pomeni pulse-width modulation, ali po slovensko pulzno širinska modulacija. Na tak način lahko z širino pulza krmilimo npr. hitrost nekega elektromotorja. PWM signal je v bistvu sestavljen iz pravokotnih impulzov konstantne frekvence. Razmerje širine impulza proti periodi imenujemo duty cikel in je izraženo v odstotkih. Če PWM signal priključimo na enosmerni motorček, lahko reguliramo njegovo hitrost vrtenja. Pri različnem duty ciklu bo tudi hitrost vrtenja enosmernega motorčka različna. Pulzno širinska modulacija v našem primeru deluje s frekvenco 500 Hz.



Slika 2: PWM Graf (vir: Arduino.cc)

Zraven tega ima Arduino še 6 analognih vhodov (posamezni vsebuje analogno digitalni pretvornik katerega ločljivost znaša 1024, zazna 1024 sprememb – od 0 do 1023) in 16 MHz kristalni oscilator, ki narekuje takt delovanja mikrokrmilnika.

Programiramo ga z brezplačnim razvojnim okoljem Arduino IDE, ki je narejeno na osnovi Java programskega jezika. Programski jezik Arduino pa temelji na jeziku Processing. Arduino komunicira z računalnikom preko USB povezave. Na tak način lahko program, napisan v Arduino programskem jeziku, naložimo na mikrokrmilnik.

Tehnične specifikacije mikrokrmilnika:

Mikrokrmilnik	ATmega328P
Napetost delovanja	5V
Vhodna napetost (priporočena)	7-12 V
Vhodna napetost (omejitve)	6-20 V
Digitalni I/O pini	14 (6 od teh je PWM)
PWM digitalni I/O pini	6
Analogni vhodni pini	6
DC tok na I/O pinu	20 mA
DC tok za 3.3V pinu	50 mA
Flash spomin	32 KB (ATmega328P) od katerih je 0.5 KB namenjeno za bootloader ¹
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Takt mikrokrmilnika	16 MHz
LED_BUILTIN ²	13
Dolžina	68.6 mm
Širina	53.4 mm
Teža	25 g

¹ Zagonski nalagalnik (ang. bootlodaer) je programska oprema, ki skrbi za zagon mikrokrmilnika. Mikrokrmilnik brez zagonskega nalagalnika ne bi deloval.

² LED dioda, ki je že vgrajena na mikrokrmilniku.

3.2 Ultrazvočni senzor HC-SR04



Slika 3: Ultrazvočni senzor HC-SR04 (vir: Google Slike)

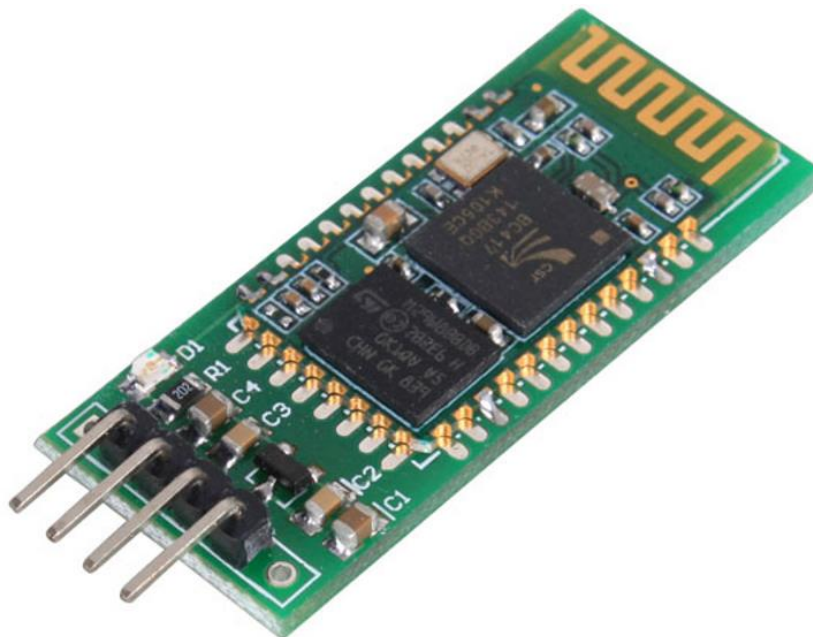
Ultrasonični oz. ultrazvočni senzor služi merjenju razdalje med robotom in oviro. Na podlagi izmerjenih podatkov lahko prepreči morebitne poškodbe na robotu, saj se pri določeni razdalji od ovire robot samodejno ustavi in je nadaljevanje vožnje možno samo v nasprotni smeri.

Najprej senzor priključimo na delovno napetost, da začne delovati. Modul meri razdaljo na podlagi časa potovanja zvoka. Preko pina, ki se imenuje »trigPin«, s pomočjo Arduina oddamo visokofrekvenčni pulz, ta pulz se odbije od predmeta in se vrne nazaj v sprejemnik in preko echoPina sprejmemo ta signal v Arduino. Glede na dobljeno časovno razliko med oddanim in sprejetim signalom izračunamo razdaljo v centimetrih.

Formula za izračun razdalje: $\frac{\mu s}{54} = \text{centimetri}$

Torej dobljeno časovno razliko v mikrosekundah moremo deliti z vrednostjo 54 in tako dobimo razdaljo v centimetrih.

3.3 Bluetooth komunikacijski modul HC-06



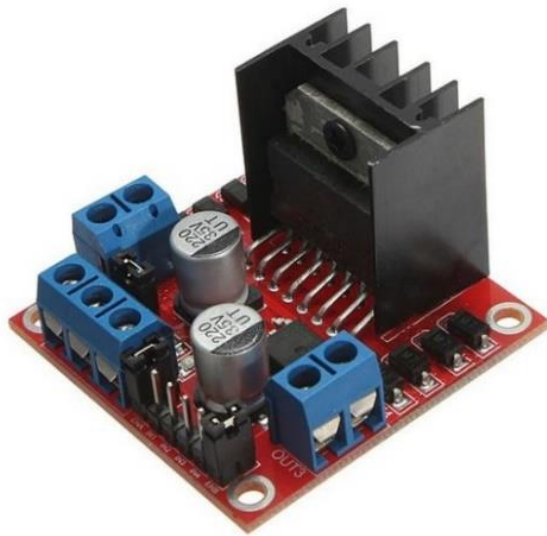
Slika 4: Bluetooth modul HC-06 (vir: Google Slike)

Bluetooth modul omogoča serijsko (zaporedno) komunikacijo med robotom in pametnim telefonom. Omogoča brezžično povezovanje med katerimikoli električnimi napravami, ki podpirajo ta standard. Bluetooth komunicira z drugimi napravami na frekvenčnem pasu okoli 2.45 GHz. Če želimo vzpostaviti povezavo z napravo Bluetooth, moramo vklopiti radijski vmesnik za Bluetooth v pametnem telefonu (Android). Ko prvič uporabimo novo napravo Bluetooth, jo moramo "združiti" z napravo, s katero jo želimo uporabljati, tako da obe napravi vesta, kako se varno povezati druga z drugo. Ko to naredimo, se samodejno povežeta.

Mi preko mobilne aplikacije, ki je naložena na pametni telefon, s pritiskom na nek gumb pošljemo serijski tok podatkov, ki ga sprejme Bluetooth modul. Tako s pomočjo sprejetih podatkov in kode krmilimo elektromotorje ali servo pogon. Priključek na plošči Arduino, ki sprejema podatke iz Bluetooth modula, se imenuje RX (receive) priključek. Modul priključimo na delovno napetost 5 V.

Hitrosti prenosa podatkov so od 1200 bps (bitov na sekundo) do 921 kbps. Maksimalna razdalja ki jo Bluetooth modul doseže pa znaša približno 20 metrov.

3.4 H-Mostič krmilnik motorjev



Slika 5: H-Mostič krmilnik motorjev (vir: Google Slike)

H-Mostič, krmilnik enosmernih motorjev, je elektronski modul, ki ga krmilimo preko Arduino mikrokrmilnika. Z majhnim krmilnim tokom s pomočjo štirih priključkov krmilimo smer vrtenja elektromotorja. Z drugima dvema krmilnima priključkoma pa s pomočjo pulzno širinske modulacije spreminjamo hitrost vrtenja elektromotorja. S tem krmilnikom se lahko ločeno krmilita dva elektromotorja. Krmilnik za delovanje potrebuje enosmerno napetost 5 V.

3.5 Servo pogon



Slika 6: Servo pogon (vir: Google Slike)

Servo je naprava, ki je sestavljena iz elektromotorja, zobniškega prenosa in elektronike. Servo motorji so samostojne električne naprave, ki premikajo in obračajo dele strojev s precizno natančnostjo. Najdemo jih skoraj povsod, od igrač do avtomobilov. Servo ima tri priključke. Dve sta za napajanje, ena pa za signal. Krmilimo ga impulzno preko signalne žice. Poleg vezic za napajanje je srce servo motorja majhen enosmerni motor, kateri se vrti pri visokih obratih. Ker ima zelo malo navora, ima zraven zobniški prenos, ki visoke obrate z malim navorom pretvori v počasne obrate z velikim navorom. Motor vsebuje še elektronsko vezje na tiskani ploščici, kjer se nahaja senzor. Ta prepozna za koliko se je zavrtela gred motorja. Pozicija gredi motorja je odvisna od dolžine impulza na priključku za signal, frekvenca teh impulzov pa je okoli 50 Hz. Ročica se iz nevtralnega položaja vrti v +90 stopinj in v -90 stopinj. Impulze pošiljamo vsakih 20 milisekund. Od širine impulza pa je odvisen kot za katerega se obrne ročica servota. Nevtralni položaj dosežemo s širino impulza 1,5 milisekunde, zasuk v -90 stopinj s širino impulza 1 milisekunde, zasuk v +90 stopinj pa s širino impulza 2 milisekundi. Za lažje krmiljenje servo pogonov smo uporabili »Servo.h« knjižnico.

3.6 DC Motor

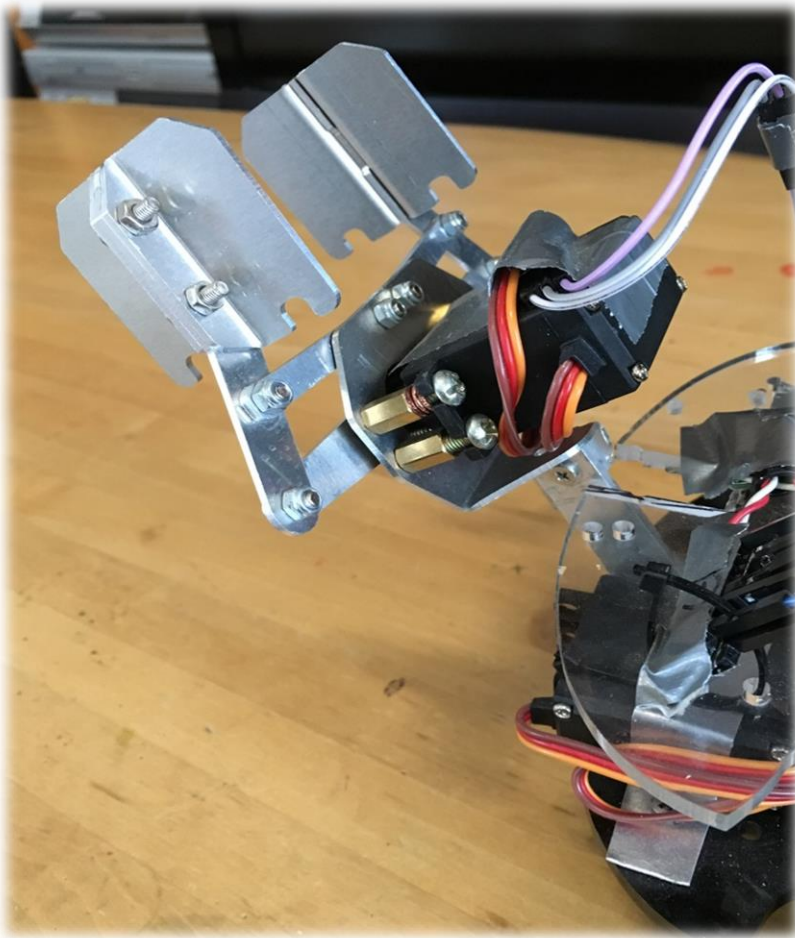


Slika 7: DC Motor (vir: Google Slike)

Motor na enosmerni tok ali DC motor je električni stroj, ki pretvarja električno energijo v mehansko. Sestavljen je iz ohišja (stator), osi (rotorja) in magnetov ter navitij. Ko skozi navitja steče tok, navitja zaradi lastnega magnetnega polja pritegnejo magnet in rotor se začne vrteti. Med magnetnimi poli statorja se nahaja rotor z navitjem, povezanim preko komutatorja in ščetk na zunanji vir enosmerne napetosti. Če menjamo polariteto priključene enosmerne napetosti, se motorček vrti v drugo smer. Za spreminjanje polaritete priključene napetosti in s tem spreminjanje smeri vrtenja elektromotorčka uporabimo H-Mostič.

Tako štirje uporabljeni elektromotorji na robotu vrtijo kolesa. Ker skozi kontakta elektromotorja teče prevelik tok, da bi motor priključili direktno na ploščo Arduino, uporabimo H-Mostič krmilnik motorjev.

3.7 Robotska roka



Slika 8: Robotska roka (vir: avtorji naloge)

Robotska roka je sestavljena iz dveh servo pogonov. Spodnji servo robotsko roko dviga ali spušča, drugi servo pogon pa preko zobnikov prestavlja robotske klešče skupaj in narazen. Robotska roka lahko prime nek predmet, ga dvigne in prestavi na željeno mesto.

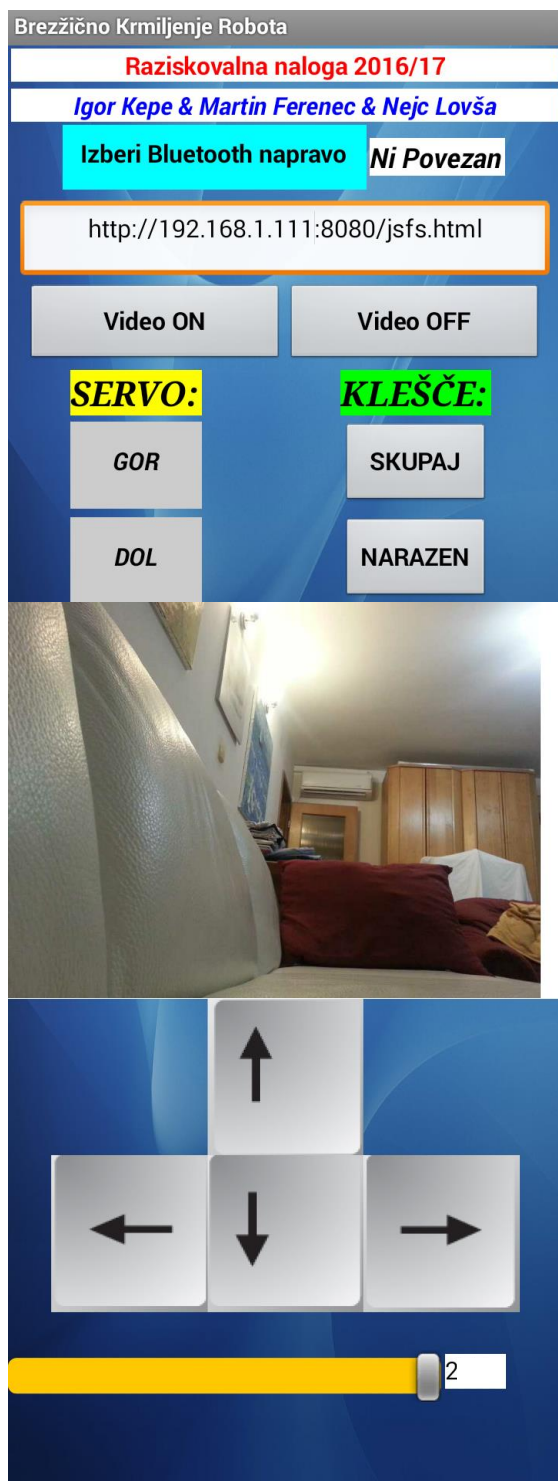
3.8 IP Kamera



Slika 9: IP Kamera (vir: avtorji naloge)

V tej raziskovalni nalogi uporabljamo tudi IP kamero. Kamera je na robotu. Omogoča pa direkten prenos videa iz kamere, preko brezžične internetne povezave (WiFi) v aplikacijo, ki je naložena na pametnem telefonu. Aplikacija IP Camera je naložen na telefonu in omogoča, da se telefon obnaša kot IP kamera.

3.9 Aplikacija



Slika 10: Aplikacija (vir: avtorji naloge)

Slika 10 prikazuje aplikacijo, ki je naložena na pametnem telefonu z operacijskim sistemom Android. Najprej s klikom na gumb »Izberi Bluetooth napravo« izberemo Bluetooth napravo in vzpostavimo povezavo med telefonom in Bluetooth modulom, ki je na Arduino. Desno se nam izpiše ali je naprava povezana, ali ne. Nato v naslednje polje vpišemo celotni lokalni IP naslov z vrati ter stisnemo gumb »Video ON« in spodaj bomo videli direkten prenos videa. Če stisnemo gumb »Video OFF«, se bo video ugasnil. To nam omogoča krmiljenje robota takrat, ko robot ni v vidnem polju. Če pod napisom »Servo« pritisnemo gumb Gor oziroma Dol, se bo robotska roka premikala gor oziroma dol, odvisno na kateri gumb pritisnemo. Če stisnemo gumba Skupaj oziroma Narazen, se bodo robotske klešče premikale skupaj oziroma narazen. S spodnjimi gumbi pa enostavno krmilimo smer vrtenja motorjev, smer gibanja robota. Pod smernimi gumbi je še drsnik, s pomočjo katerega nastavljamo hitrost gibanja robota. Desno se nam izpiše vrednost hitrosti (1 – počasi; 2 – srednje; 3 - hitro).

3.10 Programiranje in opis programske kode

Preden smo začeli s programiranjem smo uvozili knjižnico za krmiljenje servo pogonov, ki nam omogoča lažje krmiljenje pogonov. Nato smo definirali servo pogone in zaženemo serijsko komunikacijo.

```
#include <Servo.h>

Servo gordol;
Servo skupajnarazen;
Serial.begin(9600);
```

Nato smo definirali priključke za H-Mostič, krmilnik motorjev, kot digitalne izhode.

```
pinMode(I1, OUTPUT);
digitalWrite(I1, LOW);
pinMode(I2, OUTPUT);
digitalWrite(I2, LOW);
pinMode(I3, OUTPUT);
digitalWrite(I3, LOW);
pinMode(I4, OUTPUT);
digitalWrite(I4, LOW);
```

Spreminjanje hitrosti dosežemo tako, da aplikacija pošlje string »min«, »sre« ali »max«, nato pa mikrokrmilnik spremeni PWM vrednost.

```
if(t.length()>0)
{
    if (t == "min") { pwm=120;}
    if (t == "sre") { pwm=180;}
    if (t == "max") { pwm=255;}
}
```

Po enakem postopku komunikacije krmilimo tudi robota (vožnja, klešče,...).

```
if(t.length()>0)
{
    if (t == "gor" && cm > 10)
    {
        analogWrite(pwm1, pwm);
        analogWrite(pwm2, pwm);

        digitalWrite(I1, LOW);
        digitalWrite(I2, HIGH);
        digitalWrite(I3, HIGH);
        digitalWrite(I4, LOW);
    }
}
```


Premikanje robotskih klešč z uporabo servo pogonov:

Ko mikrokrmilnik dobi podatek »ss« se bodo robotske klešče premaknile za 2° skupaj. Nato pa spremenimo vrednost spremenljivke »sers« na 1. To pomeni, da ko program pride do pogojnega if stavka »če je sers = 1«, se bo servo pogon premikal do določene vrednosti (največ 180°).

```
if(t == "ss")
{
    skupajnarazen1 = skupajnarazen1 + 2;
    sers=1;

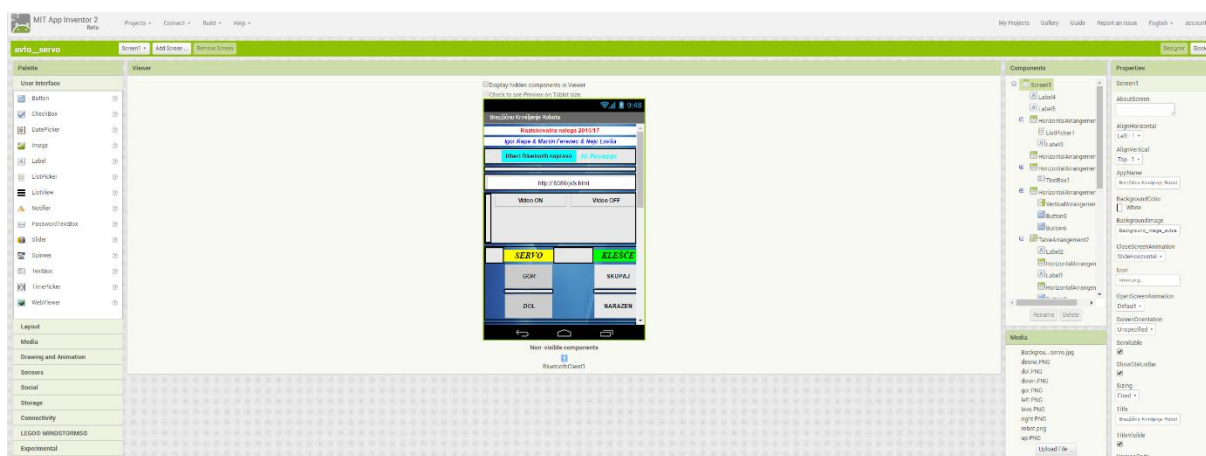
    if(gordol1 < 100)
    {
        gordol1 = 100;
    }
    if(gordol1 > 180)
    {
        gordol1 = 180;
    }
    skupajnarazen.write(skupajnarazen1);
}
```

Če pa aplikacija pošlje podatek »up«, pa se bo robot ustavil in se vse spremenljivke, ki nam pomagajo pri krmiljenju servo pogonov, spremenijo na vrednost »0«.

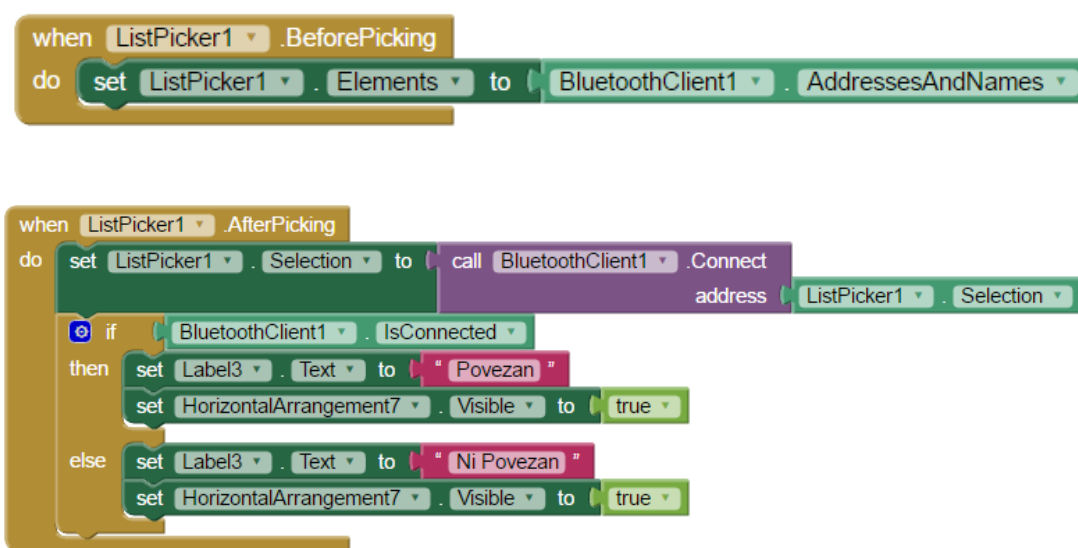
```
if(t == "up")
{
    digitalWrite(I1, LOW);
    digitalWrite(I2, LOW);
    digitalWrite(I3, LOW);
    digitalWrite(I4, LOW);
    serg=0;
    serd=0;
    sers=0;
    sern=0;
}
delay(100);
t="";
}
```

*Celotna Arduino programska koda se nahaja v prilogi, kjer je tudi komentirana.

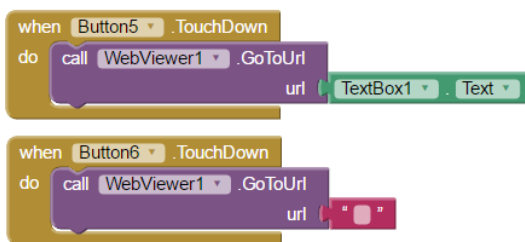
Za programiranje Android aplikacije smo uporabili MIT App Inventor 2 programsko okolje, ki je še v beta stanju (kar pomeni, da je še v fazi razvoja). MIT App Inventor 2 temelji na grafičnem programiranju.



Slika 11: Programsko okolje MIT App Inventor 2 (vir: avtorji naloge)



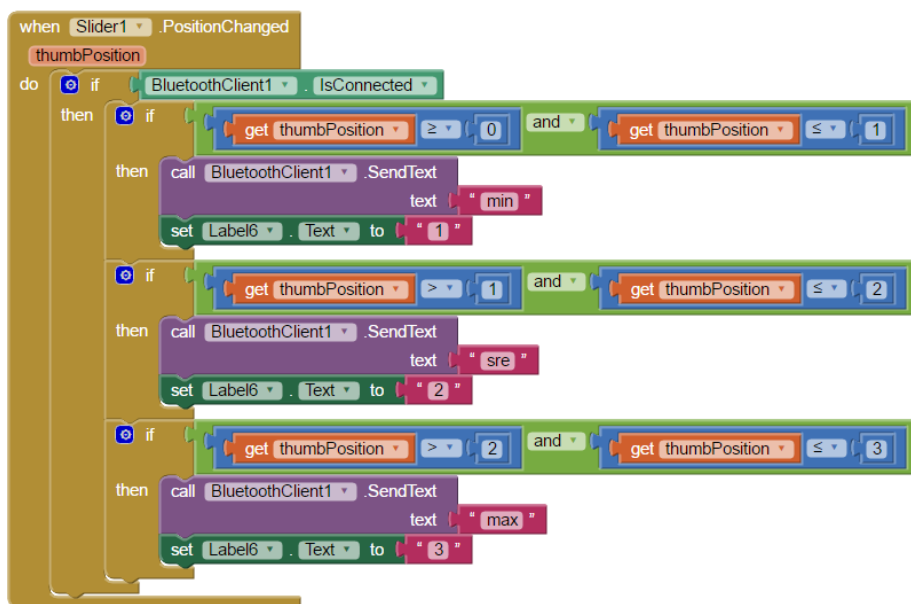
Po pritisku na gumb ListPicker1 se nam odpre novo okno, kjer izberemo Bluetooth napravo za povezovanje. Če je povezava vzpostavljena med napravo in aplikacijo, se nam v desnem okencu izpiše »Povezan« oziroma »Ni povezan«.



Ko pritisnemo »Button5«, se v »WebView1« prikaže stran (video IP kamere). URL pa dobi iz »TextBox1«.



Ko pritisku na katerikoli gumb (gor, dol, levo, desno,...) se pošlje string iz aplikacije na Bluetooth modul, ki zažene operacijo. Ko pa gumb spustimo, pošljemo ukaz za prekinitev »up«.



Zgornji blok prikazuje nastavitev hitrosti.

S pomočjo drsnika spreminjamo vrednost, ki se giblje med 1 in 3.

Vrednost 1 predstavlja najpočasnejšo hitrost premikanja robota, vrednost 3 pa najhitrejšo.

Ob tem pa se tudi spremeni številka prikazana v »Label6«, ki je ob drsniku.

*Celotna programska koda se nahaja v prilogi.

4. ZAKLJUČEK

V tej raziskovalni nalogi smo se naučili veliko o programiranju v jeziku Arduino in uporabljati grafični programator MIT App Inventor 2. Začeli smo z enostavnimi programi. Za vsak sestavni del smo naredili svoj program in ga testirali. Nato smo vse te male programe združili v enega. Potrdimo lahko vse hipoteze (zastavljene cilje). Torej sedaj deluje serijska komunikacija med telefonom in mikrokontrolerom. Vsi sestavni deli delujejo tako, kot smo si želeli.

Za izdelavo te raziskovalne naloge smo porabili veliko časa. Ni nam žal porabljenega časa, saj smo se zelo veliko naučili o programiranju in to znanje nam bo zelo koristilo v prihodnosti.

5. DRUŽBENA ODGOVORNOST

V današnjih časih si preprosto več ne predstavljamo življenja brez tehnološko naprednih naprav in rešitev. Nekaj od teh rešitev predstavlja naša raziskovalna naloga. Otroci lahko našega robota uporabljajo kot poučno igračo. V primerih, ko sumimo da je v določenem prostoru nekaj narobe, nam lahko reši življenje.

Na primer, če sumimo, da je v kurilnici preveč ogljikovega monoksida (ta je smrtno nevaren) lahko na robota namestimo senzor, ki zaznava prisotnost in količino ogljikovega monoksida v zraku. Nato pošljemo robota v kurilnico in se lahko prepričamo, če je varno vstopiti. Na robota bi lahko namestili tudi senzorje vnetljivih plinov, ki v prostoru zaznajo nevarnost eksplozije, senzor vlažnosti zraka in temperature ali pa senzor gibanja. S tem bi lahko marsikomu rešili življenje, saj je bilo veliko smrtnih žrtev, ki so se zastrupili s plini. S kamero si lahko ogledamo tudi celoten prostor in vidimo morebitne nevarnosti.

6. PRILOGE

6.1 Arduino programska koda

```
#include <Servo.h> //vkljucena je knjiznica za Servo pogone

Servo gordol;           //definiramo Servo pogon
Servo skupajnarazen;

const int trigPin = 13; //izhodni pin na ultrasonicnem senzorju
const int echoPin = 12; //vhodni pin na ultrasonicnem senzorju

int I1 = 8;
int I2 = 9;           // izhodni pini, ki krmilijo motorje (povezano z H-
Bridge)
int I3 = 10;
int I4 = 11;

int pwm;             //spremenljivka, v kateri je definirana hitrost motorja
int pwm1 = 5;        //pwm pin, s katerim krmilimo hitrost motorja
int pwm2 = 6;

int gordol1 = 180;    //Kot servo-ta, ki dviga ali spusca
int skupajnarazen1 = 90; //Kot servo-ta, ki gre skupaj in narazen
int serg;
int serd;             //spremenljivke, v kateri so shranjena stanja Servo
pogonov robotske roke
int sers;
int sern;

// Setup (nastavitveni del programa, v katerem se definirajo zacetne
vrednosti spremenljivk)-se izvede samo ob zagonu-----
-----
void setup() {

    gordol.write(180);           // servo gordol damo na pozicijo 180°
    skupajnarazen.write(90);     // servo skupajnarazen damo na pozicijo
90°
    pwm=250;           //255 je maksimalna hitrost
    serg=0;
    serd=0;           //spremenljivke damo na stanje 0
    sers=0;
    sern=0;

    Serial.begin(9600); //hitrost serijske komunikacije v baudih

    // definiramo pine kot izhode in jim dodelimo logicno stanje "0" (low)
    pinMode(I1, OUTPUT);
    pinMode(I2, OUTPUT);
    pinMode(I3, OUTPUT);
    pinMode(I4, OUTPUT);

    //servo pogonu dodelimo pin, s katerim ga krmilimo
    gordol.attach(2);
    skupajnarazen.attach(4);
}

// osnovna zanka ki se ponavlja do prekinitve-----
-----
void loop() {
```

```

    delay(10);    //program počaka 10 mili-sekund, da ne preobremenimo
procesorja

    String t; //ustvari spremenljivko t, v kateri se shranjujejo prihajajoci
podatki iz Bluetooth povezave
    while(Serial.available()) //while funkcija, ki se izvaja tako dolgo,
dokler prihajajo podatki
    {
        t += (char)Serial.read(); //prebere bajt, ga spremeni v karakter (znak)
in sestavi string t
        delay(1); // program počaka 1ms
    }

long duration, cm; // "long" spremenljivka shrani dvakrat več kot
spremenljivka int

    // senzor je sprožen s kratkim pulzom "HIGH", ki traja 10 ali več mili-
sekund
    // senzoru da kratki pulz "LOW", da zagotovimo čist "HIGH" pulz
pinMode(trigPin, OUTPUT);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

    // z drugim pinom prebere signal senzorja, ki je prišel nazaj
pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);

    // pretvori čas v distanco (centimetre)
cm = (duration /27)/2;

//ce je razdalja na ultrasoničnem senzorju manjša kot 15cm, se robot ustavi
    if(cm < 15) {
        digitalWrite(I1, LOW);
        digitalWrite(I2, LOW);
        digitalWrite(I3, LOW); //krmilne pine damo na logično stanje "0"
        digitalWrite(I4, LOW);
    }

    if(t.length()>0) { //ce string t ni prazen oz. je dolzina stringa vecja
od 0, naredi naslednje:

        if (t == "min") { pwm=120;}
        if (t == "sre") { pwm=180;} //če preko bluetooth povezave pride
"min" "sre" ali "max", priredi temu ustrezno hitrost
        if (t == "max") { pwm=255;}

```

```

if (t == "gor" && cm > 10) { //ce preko BT povezave pride "gor" in če je
distanca večja od 10cm, se robot pelje napred
    analogWrite(pwm1, pwm);
    analogWrite(pwm2, pwm);

    digitalWrite(I1, LOW);
    digitalWrite(I2, HIGH);
    digitalWrite(I3, HIGH);
    digitalWrite(I4, LOW);
}

    if (t == "dol") { //ce preko BT povezave pride "dol", se
robot pelje nazaj
    analogWrite(pwm1, pwm);
    analogWrite(pwm2, pwm);

    digitalWrite(I1, HIGH);
    digitalWrite(I2, LOW);
    digitalWrite(I3, LOW);
    digitalWrite(I4, HIGH);
}

    if(t == "levo") { //ce preko BT povezave pride "levo", se
robot pelje levo
    analogWrite(pwm1, pwm);
    analogWrite(pwm2, pwm);

    digitalWrite(I1, HIGH);
    digitalWrite(I2, LOW);
    digitalWrite(I3, HIGH);
    digitalWrite(I4, LOW);
}

    if(t == "desno") { //ce preko BT povezave pride "desno", se
robot pelje desno
    analogWrite(pwm1, pwm);
    analogWrite(pwm2, pwm);

    digitalWrite(I1, LOW);
    digitalWrite(I2, HIGH);
    digitalWrite(I3, LOW);
    digitalWrite(I4, HIGH);
}

```

```

if(t == "gs") { //ce preko BT povezave pride "gs", se servo premakne za 2°
gor in spremenljivki za krmiljenje servo-ta dodeli vrednost 1

    gordoll = gordoll + 2;
    serg=1;
    if(gordoll<80) {
        gordoll=80;
    }
    if(gordoll>180) {
        gordoll=180;
    }
    gordol.write(gordoll);
}

    if(t == "ds") { //ce preko BT povezave pride "ds", se servo premakne
za 2° dol in spremenljivki za krmiljenje servo-ta dodeli vrednost 1
    gordoll = gordoll - 2;
    serd=1;
    if(gordoll<90) {
        gordoll=90;
    }
    if(gordoll>180) {
        gordoll=180;
    }
    gordol.write(gordoll);
}

    if(t == "ss") { //ce preko BT povezave pride "ss", se servo
premakne za 2° skupaj in spremenljivki za krmiljenje servo-ta dodeli
vrednost 1
    skupajnarazen1 = skupajnarazen1 + 2;
    sers=1;
    if(gordoll<100) {
        gordoll=100;
    }
    if(gordoll>180) {
        gordoll=180;
    }
    skupajnarazen.write(skupajnarazen1);
}

    if(t == "ns") { //ce preko BT povezave pride "ns", se servo
premakne za 2° narazen in spremenljivki za krmiljenje servo-ta dodeli
vrednost 1
    skupajnarazen1 = skupajnarazen1 - 2;
    sern=1;
    if(gordoll<100) {
        gordoll=100;
    }
    if(gordoll>180) {
        gordoll=180;
    }
    skupajnarazen.write(skupajnarazen1);
}
if(t == "up") { //ce pa preko BT povezave pride "up", se robot ustavi in
spremenljivkam za krmiljenje servo-tov dodeli vrednost 0

```

```

    digitalWrite(I1, LOW);
    digitalWrite(I2, LOW);
    digitalWrite(I3, LOW);
    digitalWrite(I4, LOW);
    serg=0;
    serd=0;
    sers=0;
    sern=0;
}
delay(100); //program počaka 100 ms
t="";
}

// ko je spremenljivka "serg" 1, se v vsakem koraku servo premakne za 2
stopinji gor
if (serg == 1){
    gordol1 = gordol1 + 2;
    if(gordol1<80) {
        gordol1=80;
    }
    if(gordol1>180) {
        gordol1=180;
    }
    gordol.write(gordol1); // pove servotu naj gre v pozicijo
    spremenljivke gordol1
    delay(25); // počaka 25 ms
}

// ko je spremenljivka "serd" 1, se v vsakem koraku servo premakne za 2
stopinji dol
if (serd == 1){
    gordol1 = gordol1 - 2;
    serd=1;
    if(gordol1<80) {
        gordol1=80;
    }
    if(gordol1>180) {
        gordol1=180;
    }
    gordol.write(gordol1);
    delay(25);
}

// ko je spremenljivka "sers" 1, se v vsakem koraku servo premakne za 2
stopinji skupaj
if (sers == 1){
    skupajnarazen1 = skupajnarazen1 + 2;
    sers=1;
    if(gordol1<100) {
        gordol1=100;
    }
    if(gordol1>180) {
        gordol1=180;
    }
    skupajnarazen.write(skupajnarazen1);
    delay(40);
}

// ko je spremenljivka "sern" 1, se v vsakem koraku servo premakne za 2
stopinji narazen
if (sern == 1){
    skupajnarazen1 = skupajnarazen1 - 2;

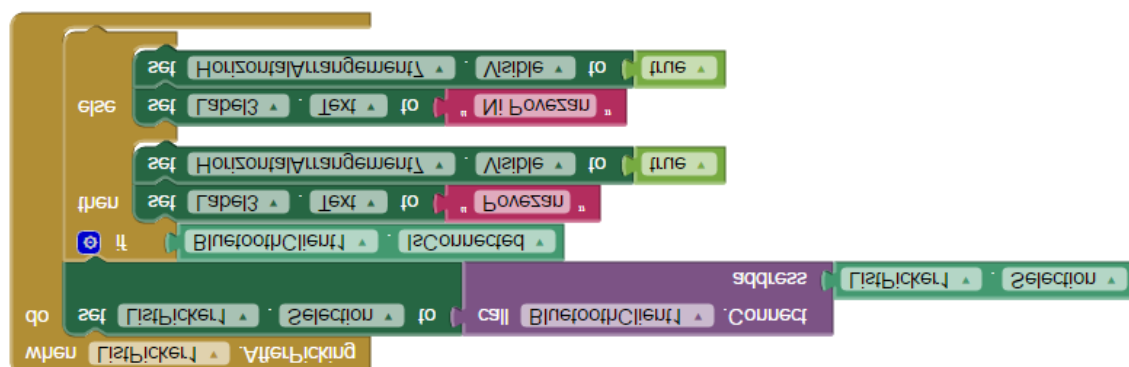
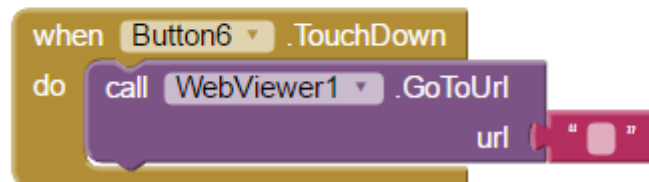
```

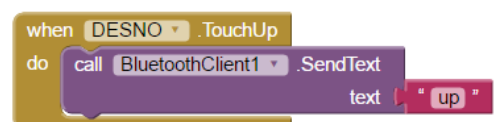
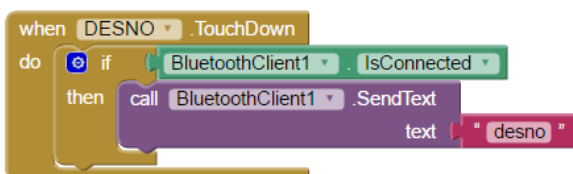
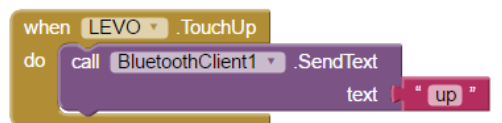
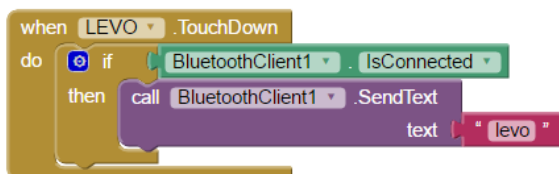
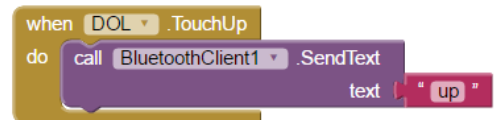
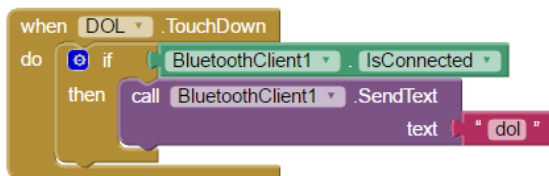
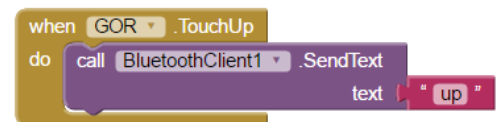
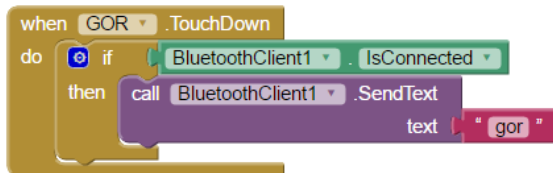
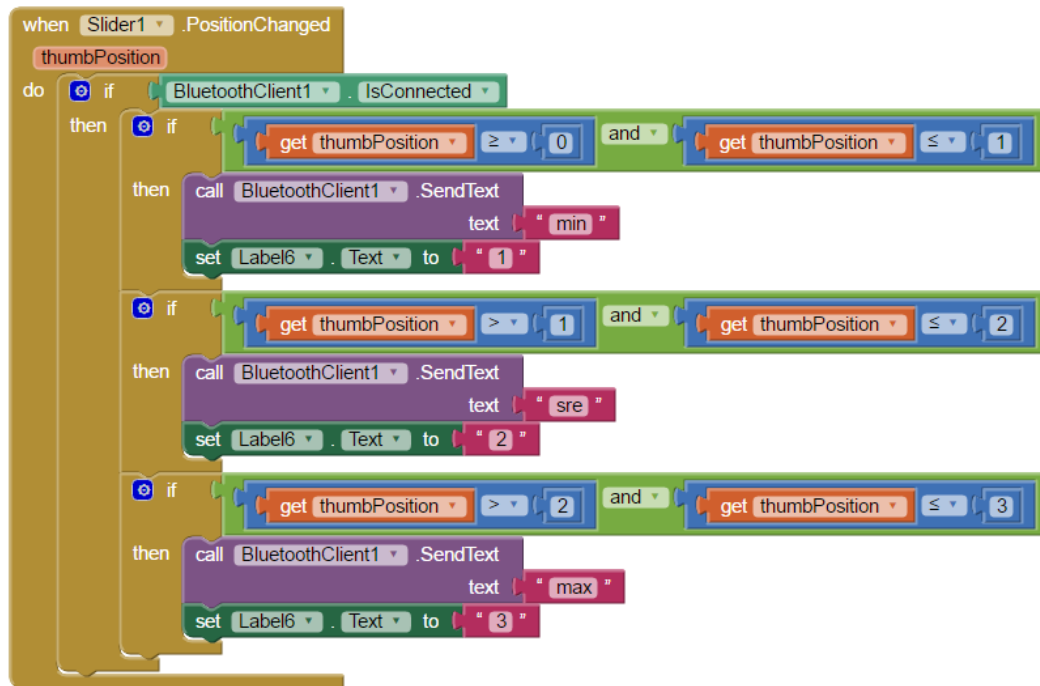
```

    sern=1;
    if(gordol1<100) {
        gordol1=100;
    }
    if(gordol1>180) {
        gordol1=180;
    }
    skupajnarazen.write(skupajnarazen1);
    delay(40);
}
}

```

6.2 MIT App Inventor 2 koda





```
when Button1 .TouchDown
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
      text "gs"
```

```
when Button2 .TouchDown
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
      text "ds"
```

```
when Button3 .TouchDown
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
      text "ss"
```

```
when Button4 .TouchDown
do
  if BluetoothClient1 .IsConnected
  then
    call BluetoothClient1 .SendText
      text "ns"
```

```
when Button1 .TouchUp
do
  call BluetoothClient1 .SendText
    text "up"
```

```
when Button2 .TouchUp
do
  call BluetoothClient1 .SendText
    text "up"
```

```
when Button3 .TouchUp
do
  call BluetoothClient1 .SendText
    text "up"
```

```
when Button4 .TouchUp
do
  call BluetoothClient1 .SendText
    text "up"
```

7. VIRI

Arduino Reference, <https://www.arduino.cc/en/Reference/HomePage> (28. januar 2017)

Arduino UNO, <https://www.arduino.cc/en/Main/ArduinoBoardUno> (28. januar 2017)

Bluetooth HC-06,

<http://fabacademy.org/archives/2015/as/students/dyatlova.polina/week14.html>

(28. januar 2017)

DC motor, <https://images-na.ssl-images-amazon.com/images/I/41TtghdA%2BpL.jpg>

(28. januar 2017)

H-Mostič krmilnik motorjev, <http://www.robotpark.com/L298N-Dual-H-Bridge-Motor-Driver> (28. januar 2017)

Programsko okolje MIT App Inventor 2, <http://appinventor.mit.edu/explore/#>

(28. januar 2017)

PWM Graf, <https://www.arduino.cc/en/Tutorial/PWM> (28. januar 2017)

Servo pogon, <https://celebratelif24x7.wordpress.com/2014/11/18/electronics-101-servo-motor/> (28. januar 2017)

Ultrasonični senzor HC-SR04, <https://robu.in/product/hc-sr04-ultrasonic-range-finder/?q=%2Fproduct%2Fhc-sr04-ultrasonic-range-finder%2F&v=ce774d9cab3a>

(28. januar 2017)