

VERSIJA

1.0

MOKOMASIS RINKINYS

QBOT

MOKOMASIS-MODULINIS ROBOTAS



electronics
ANODAS



WWW.ANODAS.LT



Mokomasis rinkinys. Versija 1.0

Parengta 2018.02.07

QBOT robotų kūrėjas ir instrukcijos rengėjas

Mindaugas Luneckas

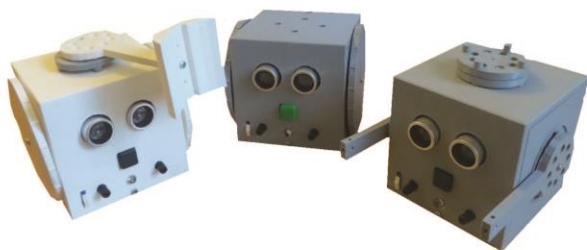
mindaugas.luneckas@gmail.com

Vilnius

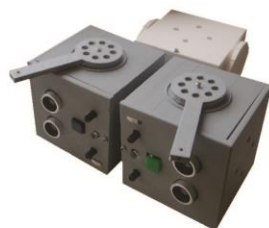
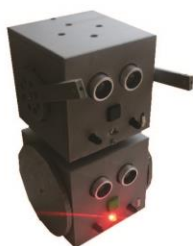
2018

TURINYS

IVADAS.....	{5}
DETALĖS RINKINYS.....	{6}
ROBOTO SURINKIMAS.....	{8}
Veido elektrinė schema.....	{9}
Galvos elektrinė schema.....	{10}
Galvos ir veido schemų jungimas.....	{12}
Komponentų montavimas ant roboto.....	{13}
Servo pavarų kalibravimas.....	{14}
Kūno ir galvos jungimas.....	{14}
Ratų montavimas.....	{15}
Jungiamosios dalys.....	{16}
ROBOTO VALDYMAS.....	{18}
Valdymo elementai.....	{19}
Darbo režimai.....	{20}
Perprogramavimas.....	{23}
Pagrindinė programa.....	{24}
PAVYZDŽIAI.....	{35}
Pavyzdys 1.....	{36}
Pavyzdys 2.....	{36}
Pavyzdys 3.....	{37}
Pavyzdys 4.....	{37}
Pavyzdys 5.....	{38}



KONSTRUOK,
PROGRAMUOK,
MĖGAUKIS!



IVADAS

QBOT: mokomasis-modulinis robotas

Žodis „robota“ išvertus iš čekų kalbos reiškia „darbas“. Atsiradus pirmiesiems robotams pasaulyje, jie būtent tam ir buvo skirti – atlikti žmogui sunkius darbus. Taigi galima sakyti, kad *robotas* – tai mašina, skirta atlikti sudėtingiems ir pavojingiems darbams. Per pastaruosius 40 metų robotai stipriai išplito visame pasaulyje ir žodžio „robotas“ prasmė šiek tiek pasikeitė. Tačiau robotų taikymas išliko toks pats. Šiandien robotai yra taikomi tokiose srityse kaip planetiniai tyrimai, gelbėjimo operacijos, išminavimo misijos, pramonės darbai ir t.t.

QBOT robotas – tai mokomoji-modulinė priemonė-konstruktorius, skirtas supažindinti su tai, kas yra robotas ir kam jie reikalingi. QBOT moko robotikos pagrindų: roboto surinkimo, programavimo ir valdymo. Tai puiki priemonė siekiant gilinti elektronikos, mechanikos žinias, o taip pat inžinerinį mąstymą, kūrybiškumą, išradingumą.

Ką reiškia *modulinis*? Modulinis reiškia, kad robotas gali būti jungiamas su kitais tokiais pačiais robotais, kad sukurti didesnius, įdomesnius robotus, kurie atliktų daug sudėtingesnius funkcijas. Tačiau tai nėra būtina; QBOT robotas gali būti naudojamas ir vienas.

QBOT *tikslas*: parodyti, kad robotai nėra taikomi vien pramogų srityse ar naudojami kaip žaislai, tačiau taip pat ir įvairiems darbams atlikti ar judėti ir tyrinėti aplinką. QBOT siekia supažindinti, kaip reikia konstruoti robotą, kad jis atliktų norimą darbą. Taip pat siekiama išmokyti, iš kokių dalių susideda robotas, kaip jis yra surenkamas, programuojamas, keičiamas ir valdomas. Darbas su robotu nereikalauja jokių bazinių įgūdžių (pvz. programavimas, fizika, matematika), nes yra pateikiamos visos sudedamosios dalys ir roboto programa. Visos turimos žinios tik palengvina darbą su QBOT robotais.

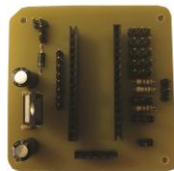
DETALĖS RINKINYJE



1 pav. Roboto kūno dalis.



2 pav. Roboto veido dalis.



3 pav. Galvos elektrinė schema.



4 pav. Veido elektrinė schema.



5 pav. Arduino Nano plokštė.



6 pav. Ultragarsinis jutiklis.



7 pav. Servo pavaros.



8 pav. AA baterijos su laikikliu.



9 pav. Spalvoti laidai.



10 pav. Kryžminis
atsuktuvus.



11 pav. Varžtai: (a) specialūs
varžtai; (b) 2x5 mm; (c) 2x10 mm.



12 pav. Jungiamosios
pavarų detalės.



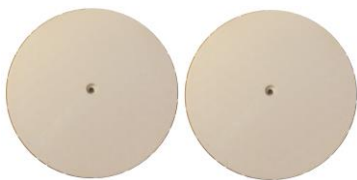
13 pav. Trumposios rankos.



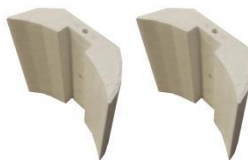
14 pav. Ilgosios rankos.



15 pav. Jungiamosios
pavarų detalės.

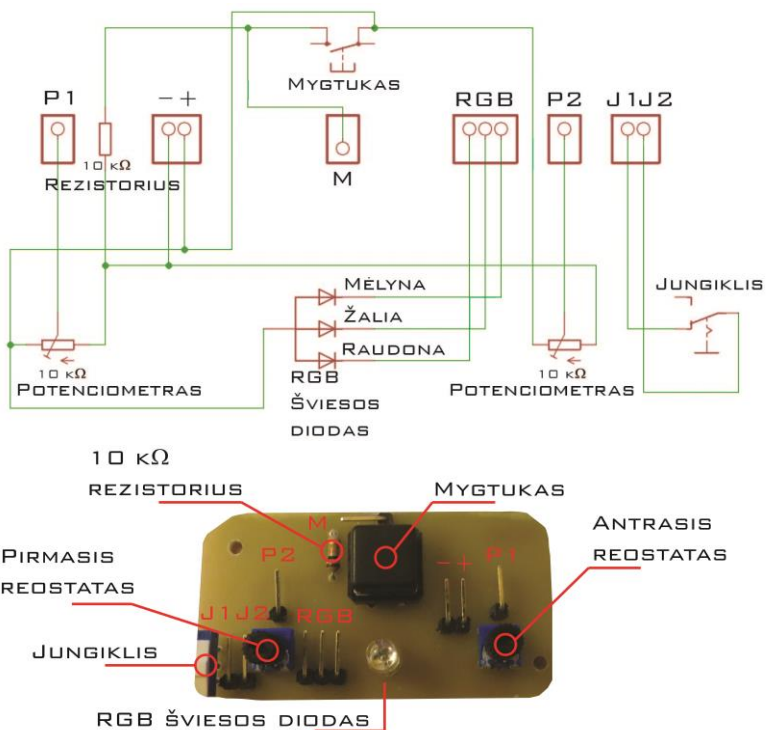


16 pav. Ratai.



17 pav. Kaušai.

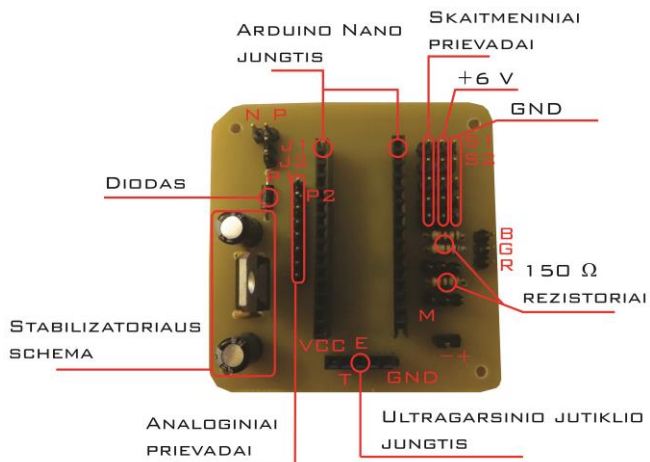
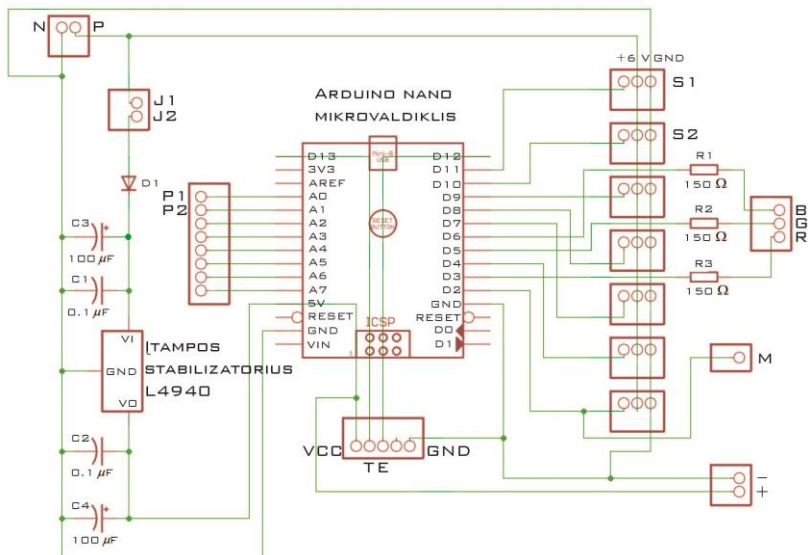
ROBOTO SURINKIMAS



Patiriamos spalvos jungtims: P1 – ruda; P2 – pilka; M – violetinė; R – oranžinė; G – žalia; B – mėlyna; + – raudona; – – juoda; J1 – balta; J2 – geltona.

Žinoma, šios spalvos gali būti laisvai pasirenkamos, nes laidai yra visiškai vienodi, nepriklausomai nuo jų spalvos.

Galvos elektrinė schema



S1 – pirmosios servo pavaros jungtis;

S2 – antrosios servo pavaros jungtis;

P1 – pirmojo potenciometro jungtis; šis potenciometras skirtas valdyti servo pavarų greitį;

P2 – antrojo potenciometro jungtis; šis potenciometras skirtas valdyti ultragarsinio reagavimo atstumą;

VCC – ultragarsinio jutiklio teigiamo poliaus jungtis;

GND – ultragarsinio jutiklio neigiamo poliaus jungtis;

T – ultragarsinio jutiklio siųstuvo (angl. trigger) jungtis;

E – ultragarsinio jutiklio imtuvo (atspindžio) jungtis (angl. echo);

P – teigiamas poliškumas (angl. positive), jungiamas nuo baterijos;

N – neigiamas poliškumas (angl. negative), jungiamas nuo baterijos;

R – raudonos šviesos diodo spalvos jungtis;

G – žalios šviesos diodo spalvos jungtis;

B – mėlynos šviesos diodo spalvos jungtis;

M – mygtuko jungtis;

J1 – pirmoji jungtiklio jungtis;

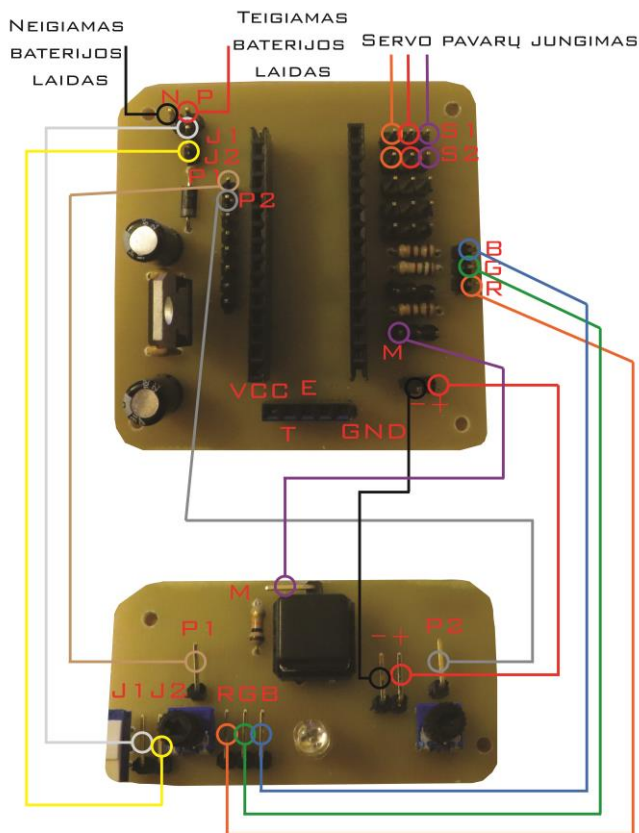
J2 – antroji jungtiklio jungtis.

+ – teigiamas maitinimo polius, skirtas perduoti maitinimą į veido schemą;

– – neigiamas maitinimo polius, skirtas perduoti maitinimą į veido schemą;

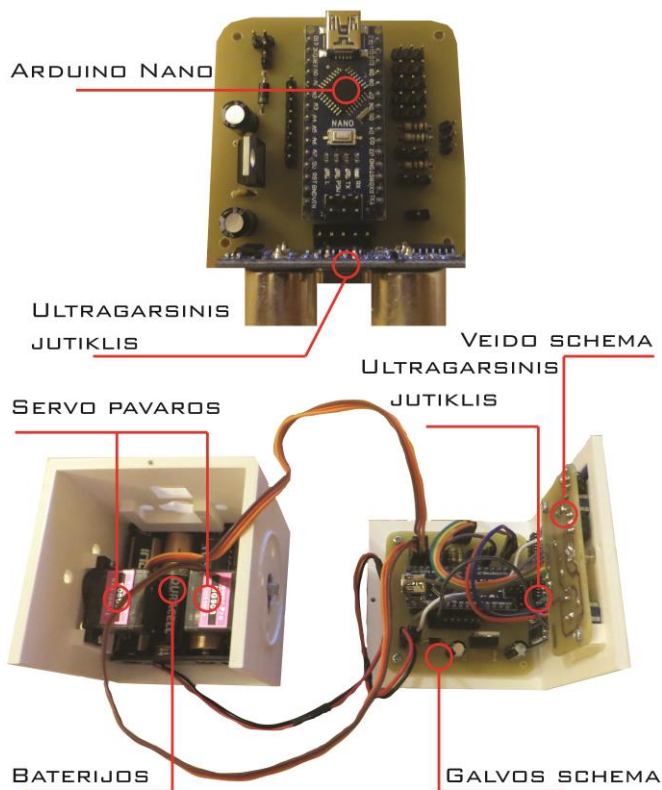
Galvos ir veido schemų jungimas

Visų pirma reikia atkreipti dėmesį į elektrines schemas, ant kurių yra sužymėtos kištukinės jungtys. Elektrinių schemų jungimas yra labai paprastas: tereikia rasti vienodus žymėjimus ant abiejų schemų ir juos sujungti. Taip daroma visoms jungtims išskyrus baterijos jungtį bei servo pavarų jungtis, kurių jungimas yra parodytas paveiksle.



Komponentų montavimas ant roboto

Toliau reikia uždėti Arduino Nano plokštę ir ultragarsinį jutiklį ant roboto galvos elektrinės schemos. Tai parodyta paveiksle apačioje. Taip pat montuojamos baterijos ir servo pavaros ant roboto kūno dalie. Servo pavaras galima montuoti trijose roboto kūno vietose, kurios pažymėtos apskritimais išorinėse sienų dalyse. Nuo to priklausys, kaip veiks robotas. Taipogi reikia sujungti servo pavarų ir baterijos laidus į roboto galvos schemą.



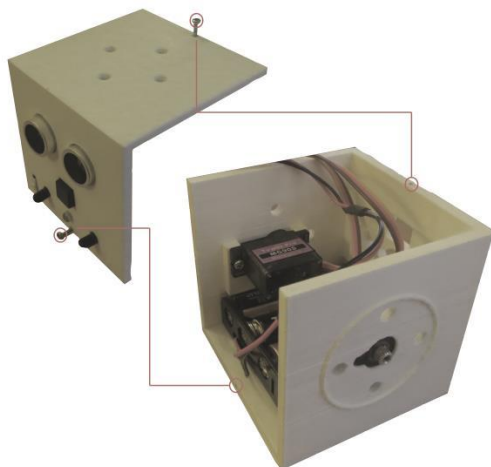
Servo pavarų kalibravimas

Servo pavarų kalibravimas yra labai paprastas. Tereikia apversti pavarą ir su kryžminiu atsuktuvu pasukti specialią svirtelę. Tai daroma įjungus pavarą, tačiau sumažinus jos greitį iki nulio. Tuo atveju, jeigu pavara vis dar sukasi, reikia sukti svirtelę tai į vieną, tai į kitą pusę tol, kol bus rasta padėtis, kai servo pavara visiškai nesisuka.



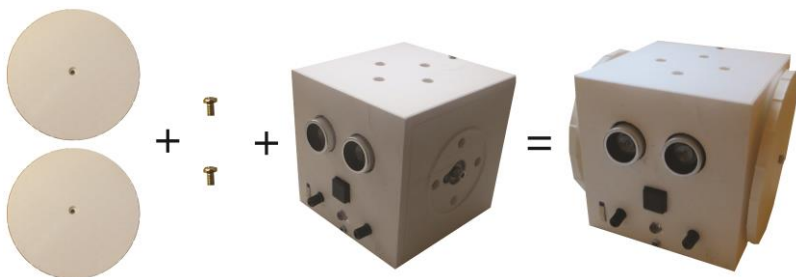
Kūno ir galvos jungimas

Sujungus roboto elektrinę dalį ir sumontavus servo pavaras bei baterijas į roboto kūną, laikas sujungti roboto galvą ir kūną. Tam reikės dviejų 10 mm ilgio varžtų ir kryžminio atsuktuvo. Roboto veido ir kūno dalys turi po dvi mažas varžtų skylutes ant sienelių kraštų. Tereikia susukti taip, kaip parodyta paveiksle ir roboto pagrindinės dalies konstravimas bus baigtas!



Ratų montavimas

Ratų montavimas taip pat nėra sudėtingas. Reikia atrasti, kuri ratų pusė turi didesnę centrinę skylę, nes šia puse ratas bus montuojamas prie servo pavaros. Tačiau reikės naudoti specialius servo pavarų varžtus. Jie yra mažiausi ir rinkinyje yra tik 2, todėl patartina jų nepamesti. Ratas uždedamas ant pavaros ir tada prisukamas varžtas taip, kad ratas nenusiimtų nuo pavaros.



Jungiamosios dalys

Kaip jau buvo minėta instrukcijos pradžioje, QBOT robotai yra darbo robotai. Jie gali būti konstruojami taip, kad atliktų įvairias užduotis, todėl nėra būtina visuomet montuoti vien ratus. Vietoj to, galima naudoti kitas rinkinyje esančias detales. Tai gali būti trumposios ir ilgosios rankos, kaušai bei jungiamosios dalys, skirtos sujungti du ar daugiau QBOT robotus. Žemiau paveiksluose pateikti detalių aprašymai bei kombinacijos, kurias galima gauti naudojant rinkinyje esančias detales.

Dėmesio! Nebūtina visas detales naudoti pagal pavadinimą. QBOT robotas yra taip pat ir konstruktorius, todėl kiekviena detalė gali būti naudojama pagal fantaziją.

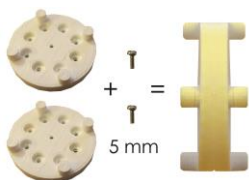
Dėmesio! Taip pat nėra būtina naudoti vien tik rinkinyje esančias detales. Galima sukurti savo unikalias dalis, pavyzdžiui, išsipjauti iš medžio ar metalo, išsikirpti iš kartono ar storo popieriaus. Tokiu būdu galima sukurti labai įvairius ir įdomius robotus, kurie atliks tam tikrą darbą.



Jungiamoji detalė. Viena tokia detalė gali būti prisukama specialiu varžtu prie servo pavaros ir tada bus galima sujungti du robotus: vieną prie pavaros, kitą – mechaniškai.



Trumpoji ir ilgoji rankos. Tokios detalės gali būti prisukamos specialiu varžtu prie servo pavaros.



Robotų jungiamoji detalė. Šia dalimi galima sujungti du QBOT robotus mechanškai.



Trumpoji ranka, kurią galima jungti ant bet kurios roboto sienos be servo pavaros. Galima jungti įvairiais kampais.



Ilgoji ranka, kurią galima jungti ant bet kurios roboto sienos be servo pavaros. Galima jungti įvairiais kampais.



Servo pavarų jungiamoji dalis. Šia dalimi galima sujungti du QBOT robotus per servo pavaras.



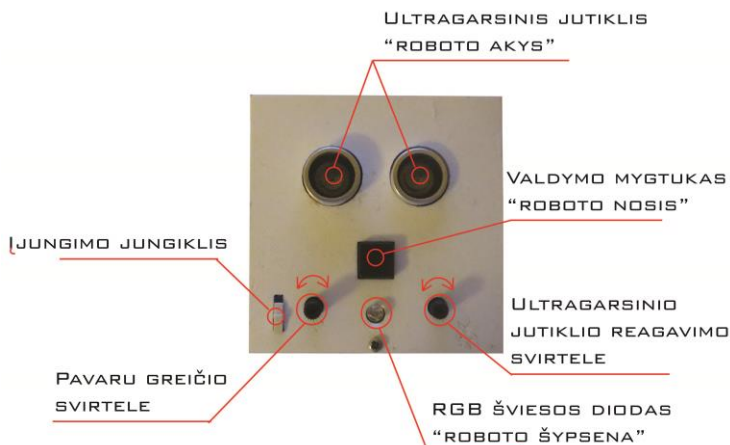
Kaušas. Šią detalę galima jungti visur, kur yra mažos skylutės. Kai kur tinka 5 mm, kai kur – 10 mm varžtai. Ši detalė yra labai universali ir gali būti naudojama kaip įvairūs roboto instrumentai.

ROBOTO VALDYMAS

Valdymo elementai

Dabar kai jau robotas yra pilnai surinktas, laikas susipažinti su roboto valdymu ir charakterio nustatymu. Visas roboto valdymas yra ant roboto veido dalies kaip pavaizduota paveiksle.

Visų pirma, robotą reikia įjungti. Tam yra skirtas įjungimo jungiklis. Pastūmus jungiklį į viršų, robotas įsijungia ir RGB šviesos diodas ima šviesti. Toliau galima apibūdinti kitus roboto veido elementus. „Roboto akys“ yra ultragarsinis jutiklis, kuris leidžia robotui aptikti priekyje esančius objektus. „Roboto nosis“ yra mygtukas, kurį paspaudus keičiasi roboto darbo režimas. „Roboto šypsena“ yra RGB šviesos diodas, kurio spalva parodo roboto „nuotaiką“ ir darbo režimą. RGB šviesos diodo abiejuose šonuose yra po vieną potenciometrų svirtelę. Viena iš jų yra skirta keisti servo pavarų greičiui, o kita – ultragarsinio jutiklio reagavimo atstumui.



Dėmesio! Greičio ir ultragarsinio jutiklio svirtelės gali būti sukeistos vietomis. Abu potenciometrai yra vienodi, todėl nėra skirtumo, kaip juos sujungti.

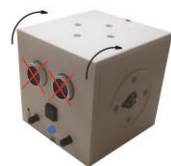
Darbo režimai

QBOT robotas pagrindinėje programoje turi užprogramuotus net 13 skirtingų darbo režimų, kurie gali būti pritaikyti kuriant skirtingus roboto elgesius. Norint perjungti skirtingus režimus, tereikia paspausti roboto nosį. Kiekvienas režimas skiriasi servo pavarų sukimosi kryptimis, o režimų kategorijos skiriasi ir spalva. Režimų kategorijos yra tokios: neutralus režimas – mėlynas (M), sekimo režimas – raudonas (R), traukimosi režimas – žalias (Ž), rankų režimas – geltonas (G), savarankiškumo režimas – violetinis (V). Visi režimai užprogramuoti iš eilės nuo 0 iki 12. Toliau pateikti visų darbo režimų aprašymai.

Dėmesio! Patartina režimus keisti nustačius minimalų pavarų greitį, nes tuo atveju, jei bus tarpusavyje sujungti keli robotai, ar prijungtos papildomos detalės, neišvys gedimas.



M0 – ultragarsinis jutiklis neveikia; abi roboto pavaros sukasi į priekį nustatytu greičiu.



M1 – ultragarsinis jutiklis neveikia; abi roboto pavaros sukasi atgal nustatytu greičiu.



M2 – ultragarsinis jutiklis neveikia; roboto pavaros sukasi skirtingomis kryptimis nustatytu greičiu.



R3 – roboto pavaros sukasi pirmyn nustatytu greičiu, kliūčiai esant mažesniu nei nustatytas atstumas.



R4 – roboto pavaros sukasi pirmyn nustatytu greičiu, iki kol nėra kliūtis atstumu, mažesniu nei nustatytas.



Ž5 – roboto pavaros sukasi atgal nustatytu greičiu, kliūčiai esant mažesniu nei nustatytas atstumas.



Ž6 – roboto pavaros sukasi atgal nustatytu greičiu, iki kol nėra kliūtis atstumu, mažesniu nei nustatytas.



G7 – ultragarsinis jutiklis neveikia; roboto pavaros sukasi pirmyn ir atgal nustatytu greičiu.



G8 – ultragarsinis jutiklis neveikia; roboto pavaros sukasi pirmyn ir atgal nustatytu greičiu skirtingomis kryptimis.



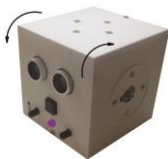
G9 – roboto pavaros sukasi pirmyn ir atgal nustatytu greičiu, kliūčiai esant mažesniu atstumu nei nustatytas.



G10 – roboto pavaros sukasi pirmyn ir atgal nustatyto greičiu, kliūčiai esant mažesniu atstumu nei nustatytas skirtingomis kryptimis.



V11 – kliūčių išvengimo režimas; geriausiai tinka su ratais. Robotas važiuoja į priekį nustatyto greičiu, kol nėra kliūtis mažesniu nei nustatytas atstumu. Atsiradus kliūčiai, robotas sustoja, pavažiuoja atgal, pasisuka ir važiuoja toliau.

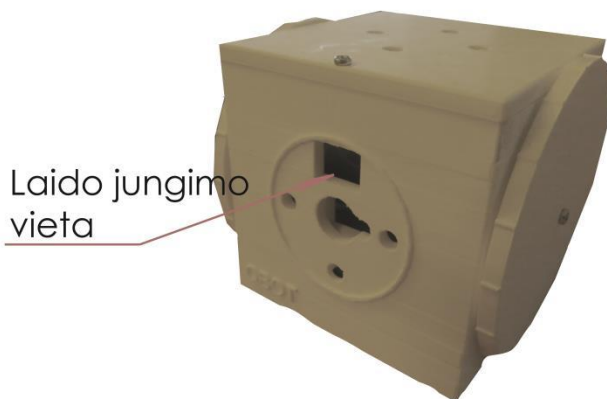


V12 – stumdymo režimas („sumo“); geriausiai tinka su ratais. Robotas sukasi vietoje nustatyto greičiu, iki kol nėra kliūtis nustatyto atstumu. Atsiradus kliūčiai, robotas pradeda judėti link kliūtis, ją kurį laiką stumia, o tada važiuoja atgal ir vėl pradeda sukstis.

Perprogramavimas

Norint pakeisti QBOT roboto veikimą, juos taip pat galima perprogramuoti. Kadangi naudojamas Arduino mikrovaldiklis, nesunku prisijungti laidu prie roboto ir įkelti savo sukurtą programą. Tereikia paimti usb laidą ir įjungti jį pro specialią skylę, esančią roboto gale. Norint rašyti programą, reikia atsisiųsti specialų arduino kompiliatorių iš arduino.cc internetinės svetainės skiltyje „software“. Taip pat galima programuoti robotą jo nesurinkus. QBOT elektrinės schemos sukurtos taip, kad jas galima būtų naudoti ne tik šiems robotams, bet ir kuriant savo projektus.

Dėmesio! Perprogramavus robotą bus ištrinti jo pagrindinė programa, kurią vėliau bus sunku perrašyti iš naujo. Siūloma perprogramuoti robotą tik turint daugiau programavimo patirties.



Pagrindinė programa

//Visą programą (įskaitant ir šią eilutę), galima kopijuoti tiesiai į Arduino programą ir ji veiks be papildomo programavimo.

#include <Servo.h> //Įkeliama servo pavarų biblioteka.

//Nustatomi programoje naudojami kintamieji ir jų pradinės reikšmės.

```
Servo pavara_1, pavara_2;
const int pot_1 = A0, pot_2 = A1;
const int button = 2;
const int trig = 13, echo = 12;
const int RED = 3, GREEN = 5, BLUE = 6;
int type = 0, brightness, button_delay = 0;
int pot1_value, pot2_value, motor_delay = 0, pos = 0;
int pavara1_speed, pavara2_speed, duration, distance, reaction;
bool avoiding = false, pushing = false, catching = false;
```

//Kintamųjų konfigūravimo funkcija. Ji atliekama vieną kartą.

```
void setup()
{
    pavara_1.attach(10);
    pavara_2.attach(11);
    pinMode(pot_1, INPUT);
    pinMode(pot_2, INPUT);
    pinMode(button, INPUT);
    pinMode(echo, INPUT);
    pinMode(RED, OUTPUT);
    pinMode(GREEN, OUTPUT);
    pinMode(BLUE, OUTPUT);
    pinMode(trig, OUTPUT);
    Serial.begin(9600);
}
```

//Pagrindinis programos ciklas. Čia sukeliamos visos robotui reikalingos funkcijos.

```
void loop()
{
    changing_type(); //Darbo režimo keitimo funkcija (nosies spaudinėjimas).
    read_values(); //Potenciometrų nuskaitymo funkcija.
    set_color(); //RGB šviesos diodo spalvos nustatymo funkcija.
    set_reaction(); //Ultragarinio jutiklio reagavimo atstumo nustatymo funkcija.
```



```
set_motor_speed(); //Pavarų sukimosi krypčių ir greičių nustatymo funkcija.  
delay(10);  
}
```

// Darbo režimo keitimo funkcija.

```
void changing_type()  
{  
  if(digitalRead(button) == 1 && button_delay > 50)  
  {  
    if(type == 13)  
      type = 0;  
    else type++;  
    button_delay = 0;  
    analogWrite(RED, 255);  
    analogWrite(GREEN, 255);  
    analogWrite(BLUE, 255);  
    delay(100);  
  }  
  if(type == 13)  
    type = 0;  
  button_delay++;  
}
```

//Potenciometrų nuskaitymo funkcija.

```
void read_values()  
{  
  pot1_value = analogRead(pot_1);  
  pot2_value = analogRead(pot_2);  
}
```

//RGB šviesos diodo spalvos nustatymo funkcija.

```
void set_color()  
{  
  if(type == 0 || type == 1 || type == 2) //Mėlyna spalva  
  {  
    //Neutrali būseną; robotas nereaguoja į kliūtis; režimai M0, M1 ir M2.  
    brightness = map(pot1_value, 0, 1023, 255, 0);  
    analogWrite(RED, 255);  
    analogWrite(GREEN, 255);  
    analogWrite(BLUE, brightness);  
  }  
  else if(type == 3 || type == 4) //Raudona spalva.
```

```

{
    //Pikta būsena; robotas seka paskui kliūtis; režimai R3 ir R4.
    brightness = map(pot1_value, 0, 1023, 255, 0);
    analogWrite(RED, brightness);
    analogWrite(GREEN, 255);
    analogWrite(BLUE, 255);
}
else if(type == 5 || type == 6) //Žalia spalva.
{
    //Draugiška būsena; robotas traukiasi nuo kliūčių; režimai Z5 ir Z6.
    brightness = map(pot1_value, 0, 1023, 255, 0);
    analogWrite(RED, 255);
    analogWrite(GREEN, brightness);
    analogWrite(BLUE, 255);
}
else if(type == 7 || type == 8 || type == 9 || type == 10) //Geltona spalva.
{
    //Darbo būsena; pavaros sukasi aukštyn žemyn; režimai G7, G8, G9 ir G10.
    brightness = map(pot1_value, 0, 1023, 255, 0);
    analogWrite(RED, brightness);
    analogWrite(GREEN, brightness);
    analogWrite(BLUE, 255);
}
else if(type == 11 || type == 12) //Violetinė spalva.
{
    //Savarankiška būsena; robotas važinėja ir išvengia kliūčių; režimai V11 ir V12.
    brightness = map(pot1_value, 0, 1023, 255, 0);
    analogWrite(RED, brightness);
    analogWrite(GREEN, 255);
    analogWrite(BLUE, brightness);
}
}

//Pavarų sukimosi krypčių ir greičių nustatymo funkcija.
void set_motor_speed()
{
    //Pavaros sukasi į priekį. Režimas M0.
    if(type == 0)
    {
        pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
        pavara_1.write(pavara1_speed);
    }
}

```

```

pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
pavara_2.write(pavara2_speed);
}
//Pavaros sukasi atgal. Režimas M1.
else if(type == 1)
{
pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
pavara_1.write(pavara1_speed);
pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
pavara_2.write(pavara2_speed);
}
//Pavaros sukasi į skirtingas puses. Režimas M2.
else if(type == 2)
{
pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
pavara_1.write(pavara1_speed);
pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
pavara_2.write(pavara2_speed);
}
//Pavaros sukasi į priekį, jei yra kliūtis mažesniu nei nustatytas atstumu.
Režimas R3.
else if(type == 3)
{
if(ultrasonic() < reaction)
{
pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
pavara_1.write(pavara1_speed);
pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
pavara_2.write(pavara2_speed);
}
else
{
pavara_1.write(90);
pavara_2.write(90);
}
}
//Pavaros sukasi į priekį tol, kol yra kliūtis mažesniu nei nustatytas atstumu.
Režimas R4.
else if(type == 4)
{
if(ultrasonic() < reaction)
{

```

```

    pavara_1.write(90);
    pavara_2.write(90);
}
else
{
    pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
    pavara_1.write(pavara1_speed);
    pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
    pavara_2.write(pavara2_speed);
}
}
//Pavaros sukasi atgal, jei yra kliūtis mažesniu nei nustatytas atstumu. Režimas
Z5.
else if(type == 5)
{
    if(ultrasonic() < reaction)
    {
        pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
        pavara_1.write(pavara1_speed);
        pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
        pavara_2.write(pavara2_speed);
    }
    else
    {
        pavara_1.write(90);
        pavara_2.write(90);
    }
}
//Pavaros sukasi atgal tol, kol yra kliūtis mažesniu nei nustatytas atstumu.
Režimas Z6.
else if(type == 6)
{
    if(ultrasonic() < reaction)
    {
        pavara_1.write(90);
        pavara_2.write(90);
    }
    else
    {
        pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
        pavara_1.write(pavara1_speed);
        pavara2_speed = map(pot2_value, 0, 1023, 45, 90);

```

```

    pavara_2.write(pavara2_speed);
}
}
//Pavaros sukasi pirmyn atgal vienodomis kryptimis; jutiklis neveikia. Režimas
G7.
else if(type == 7)
{
    if(pos < 30)
    {
        pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
        pavara_1.write(pavara1_speed);
        pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
        pavara_2.write(pavara2_speed);
    }
    else if(pos > 30 && pos < 60)
    {
        pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
        pavara_1.write(pavara1_speed);
        pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
        pavara_2.write(pavara2_speed);
    }
    if(pos >= 60)
        pos = 0;
    else pos++;
}
//Pavaros sukasi pirmyn atgal vienodomis kryptimis, jei yra klūtis mažesniu
nei nustatytas atstumu. Režimas G8.
else if(type == 8)
{
    if(ultrasonic() < reaction)
    {
        if(pos < 30)
        {
            pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
            pavara_1.write(pavara1_speed);
            pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
            pavara_2.write(pavara2_speed);
        }
        else if(pos > 30 && pos < 60)
        {
            pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
            pavara_1.write(pavara1_speed);

```

```

    pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
    pavara_2.write(pavara2_speed);
}
if(pos >= 60)
    pos = 0;
else pos++;
}
else
{
    pavara_1.write(90);
    pavara_2.write(90);
}
}
//Pavaros sukasi pirmyn atgal skirtingomis kryptimis; jutiklis neveikia.
Režimas G9.
else if(type == 9)
{
    if(pos < 30)
    {
        pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
        pavara_1.write(pavara1_speed);
        pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
        pavara_2.write(pavara2_speed);
    }
    else if(pos > 30 && pos < 60)
    {
        pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
        pavara_1.write(pavara1_speed);
        pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
        pavara_2.write(pavara2_speed);
    }
    if(pos >= 60)
        pos = 0;
    else pos++;
}
//Pavaros sukasi pirmyn atgal skirtingomis kryptimis, jei yra kliūtis mažesniu
nei nustatytas atstumu. Režimas G10.
else if(type == 10)
{
    if(ultrasonic() < reaction)
    {
        if(pos < 30)

```

```

{
    pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
    pavara_1.write(pavara1_speed);
    pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
    pavara_2.write(pavara2_speed);
}
else if(pos > 30 && pos < 60)
{
    pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
    pavara_1.write(pavara1_speed);
    pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
    pavara_2.write(pavara2_speed);
}
if(pos >= 60)
    pos = 0;
else pos++;
}
else
{
    pavara_1.write(90);
    pavara_2.write(90);
}
}

//V11 darbo režimas. Tinka labiausiai robotui su ratais. Robotas važinėja ir
išvengia kliūčių. Robotas važiuoja tiesiai, jei nėra priekyje kliūčių mažesniu nei
nustatytas atstumu. Atsiradus kliūčiai, robotas sustoja, pasisuka į šoną ir vėl
važiuoja tiesiai.
else if(type == 11)
{
    if(ultrasonic() < reaction)
        avoiding = true;

    if(avoiding == true)
    {
        if(pos < 20)
        {
            pavara_1.write(90);
            pavara_2.write(90);
        }
        else if(pos > 20 && pos < 40)
        {
            pavara1_speed = map(pot2_value, 0, 1023, 120, 90);

```

```

    pavara_1.write(pavara1_speed);
    pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
    pavara_2.write(pavara2_speed);
}
else if(pos > 40 && pos < 60)
{
    pavara_1.write(90);
    pavara_2.write(90);
}
else if(pos > 60 && pos < 80)
{
    pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
    pavara_1.write(pavara1_speed);
    pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
    pavara_2.write(pavara2_speed);
}
else if(pos > 80 && pos < 100)
{
    pavara_1.write(90);
    pavara_2.write(90);
}

if(pos >= 100)
{
    pos = 0;
    avoiding = false;
} else pos++;
}
else if(avoiding == false)
{
    pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
    pavara_1.write(pavara1_speed);
    pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
    pavara_2.write(pavara2_speed);
}
}

```

//V12 darbo režimas. Tinka labiausiai robotui su ratais. Dažniausiai žinomas kaip „sumo režimas“. Robotas sukasi vienoje vietoje tol, kol priekyje nemato kliūties. Atsiradus kliūčiai robotas pradeda važiuoti į priekį tol, kol nustumia priekyje esantį objektą nedideliu atstumu į priekį. Tada pavažiuoja atgal ir vėl pradeda sukis vietoje.

```

else if(type == 12)

```



```

{
  if(ultrasonic() < reaction)
    catching = true;

  if(catching == true)
  {
    if(ultrasonic() <= 5)
    {
      pushing = true;
      catching = false;
    }
    else if(ultrasonic() > 5)
    {
      pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
      pavara_1.write(pavara1_speed);
      pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
      pavara_2.write(pavara2_speed);
    }
  }
  if(pushing == true)
  {
    if(pos < 100)
    {
      pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
      pavara_1.write(pavara1_speed);
      pavara2_speed = map(pot2_value, 0, 1023, 120, 90);
      pavara_2.write(pavara2_speed);
    }
    else if(pos >= 100 && pos < 200)
    {
      pavara1_speed = map(pot2_value, 0, 1023, 120, 90);
      pavara_1.write(pavara1_speed);
      pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
      pavara_2.write(pavara2_speed);
    }
    else if(pos >= 200 && pos < 230)
    {
      pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
      pavara_1.write(pavara1_speed);
      pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
      pavara_2.write(pavara2_speed);
    }
  }
}

```

```

    if(pos >= 230)
    {
        pos = 0;
        pushing = false;
    } else pos++;
}
else
{
    pavara1_speed = map(pot2_value, 0, 1023, 45, 90);
    pavara_1.write(pavara1_speed);
    pavara2_speed = map(pot2_value, 0, 1023, 45, 90);
    pavara_2.write(pavara2_speed);
}
}
}

```

//Ultragarsinio jutiklio reagavimo atstumo nustatymo funkcija.

```

void set_reaction()
{
    if(pot1_value <= 300)
        reaction = 5;
    else if (pot1_value > 300 && pot1_value <= 600)
        reaction = 20;
    else if (pot1_value > 600 && pot1_value <= 900)
        reaction = 35;
    else if (pot1_value > 900 && pot1_value <= 1023)
        reaction = 50;
}

```

//Ultragarsinio jutiklio nuskaitymo funkcija. Išduodamas atstumas centimetrais.

```

int ultrasonic()
{
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);
    duration = pulseIn(echo, HIGH);
    distance = duration*0.034/2;
    return distance;
}

```

PAVYZDŽIAI

Pavyzdys: didelių objektų stumdytojas



Šis robotas gali važinėti ir stumdyti priekyje esančius didelius objektus. Naudojamas vienas QBOT robotas su dviem servo pavaromis iš šonų ir ratais. Ant viršiaus montuojama ilgoji ranka su prisuktu kaušu. Tinkamiausias darbo režimas: V12, tačiau tinka ir M0 bei R3 režimai.

2 pavyzdys: mažų objektų stumdytojas



Mažų detalių stumdytojas yra važiuojantis QBOT robotas su dviem servo pavaromis ir ratais. Robot gale mechaniškai pritvirtinta trumpoji ranka ant kurios statmenai prisuktas kaušas. Šis robotas gali stumdyti mažus objektus, kurie yra gale roboto. Tinkamiausias darbo režimai: M1, Ž5 arba Ž6. Naudojant kitus režimus robotas tiesiog neatliks jokio darbo.

3 pavyzdys: ekskavatorius



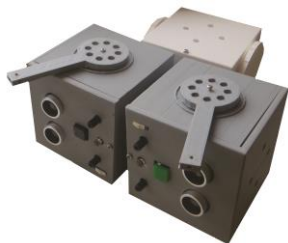
Robotas ekskavatorius gali važinėti ir kilnoti priekyje esančius objektus ar kasti smėlį. Šiam robotui sukonstruoti reikalingi 2 QBOT robotai. Vienas robotas konstruojamas su dviem servo pavaromis ir ratais, o prie jo galinės sienos jungiamas kitas robotas su dviem ilgosiomis rankomis, primontuotų ant servo pavarų. Ant ilgųjų rankų prisukti du kaušai. Važiuojančio roboto darbo režimai gali būti šie: M0, Ž5 ar Ž6. Roboto su kaušais tinkamiausi darbo režimai yra G7 arba G9.

4 pavyzdys: griovėjas



Robotas griovėjas gali važinėti ir griauti priekyje esančius įvairius objektus. Toks robotas gali imituoti pastatų griovimą. Šiam robotui sukurti reikia dviejų QBOT robotų. Apatinis robotas yra važiuojantis, kuriam tinkamiausi darbo režimai yra M0, M2, R3, G7 ar V12. Viršutinis robotas konstruojamas su dviem ilgosiomis rankomis, o tinkamiausi darbo režimai yra G7, G8, G9 ar G10.

5 pavyzdys: krabas



Robotas krabas imituoja bio-robotikos sritį, kai kuriami robotai pagal gamtoje esančius gyvius (vabzdžius ar gyvūnus). Toks robotas gali važinėti bei sugriebti priekyje esančius objektus. Robotui krabui sukurti reikalingi trys QBOT robotai. Galinis robotas yra važiuojantis, kuriam tinka šie darbo režimai: M1, M2, Ž5 arba Ž6. Priekyje esantys du robotai turi po vieną servo pavarą, ant kurių primontuotos ilgosios rankos, imituojančios gnybtus. Šiems robotams tinkamiausi darbo režimai yra G7, G8, G9 arba G10.

Elektrinių schemų sudarymui naudota programa *Eagle*:

<https://www.autodesk.com/products/eagle/overview>

Robotų valdymo programa sukurta naudojant Arduino
programavimo kalbą:

<https://www.arduino.cc/en/Main/Software>

Dėl pirkimo ir užsakymų
kreiptis elektroniniu paštu:
info@anodas.lt

Atsiradus klausimams ar gedimams,
kreiptis elektroniniu paštu:
mindaugas.luneckas@gmail.com

