



## INFORMATIKOS FAKULTETAS

**T125B158 Robotų programavimo technologijos**

**Antras laboratorinis darbas**

Studentai: Eligijus Kiudys IFF-7/14  
Martynas Girdžiūna IFF-7/14  
Simonas Brasas IFF-7/9

Dėstytojai: Doc. Narbutaitė Lina  
Doc. Adomkus Tomas

KAUNAS 2020

# Turinys

1. Tikslas, užduotis, komanda .....	3
2. Užduotys .....	3
2.1. Robotų režimai .....	3
2.2. Uždaros trajektorijos apvažiavimas ir kliūties išvengimas (Veikimas parodytas) .....	4
2.2.1. Užduoties aprašymas .....	4
2.2.2. Užduoties atlikimas .....	4
2.2.3. Dalinė išvada .....	6
2.3. Parkavimo Sistema (Veikimas neparodytas) .....	7
2.3.1. Užduoties aprašymas .....	7
2.3.2. Užduoties atlikimas .....	7
2.3.3. Dalinė išvada .....	9
2.4. Estafetes pradavimas (Veikimas neparodytas) .....	10
2.4.1. Užduoties aprašymas .....	10
2.4.2. Užduoties atlikimas .....	10
2.4.3. Dalinė išvada .....	12
3. Išvados .....	13

## Paveikslų sąrašas

Pav. 1 Uždaros trajektorijos apvažiavimo algoritmas .....	4
pav. 2 Argumentų parinkimas .....	5
pav. 3 Linijos sekimo algoritmas .....	5
pav. 4 Kliūties išvengimo algoritmas .....	6
pav. 5 Kliūties privažiavimas ir pranešimo gavimas .....	7
pav. 6 Pranešimo laukimas ir būsenų pakeitimas .....	8
Pav. 7 pranešimų siuntimas .....	9
Pav. 8 Estafetės perdavimo pirmojo roboto algoritmas .....	10
pav. 9 Estafetės perdavimo antrojo roboto algoritmas .....	11
pav. 10 onBoard roboto įjungimas .....	11
pav. 11 Pirmojo roboto važiavimo algoritmas iki antrojo roboto ir žinutės išsiuntimas .....	11
pav. 12 Antrojo roboto važiavimo algoritmas gavus žinutę ir perdavimas žinutę atgal pirmajam .....	12

# 1. Tikslas, užduotis, komanda

Susipažinti su roboto „mBot“ linijos sekimo sensoriaus veikimu. Išanalizuoti mBot IR (infraraudonų spindulių) perduodamos informacijos veikimą.

Darbas buvo atliekamas naudojant vieną kompiuterį grupiškai, t.y. programavimas, analizavimas ir ataskaitos darymas.

## 2. Užduotys

- Robotų paleidimas ir sustabdymas vykdomas IR pulteliu.
- Robotas dirba keliais režimais.

### 2.1. Robotų režimai

1. Uždaros trajektorijos apvažiavimas ir kliūties išvengimas (Veikimas parodytas)
2. Parkavimo sistema (Veikimas neparodytas)
3. Estafetes perdavimas (Veikimas neparodytas)

Parkavimo sistema buvo padaryta CoppeliaSim aplinkoje. Sistemos veikimo vaizdo įrašas:

<https://youtu.be/JDTDXpBXqC4>

Naudotas parkavimo sistemos robotas CoppeliaSim aplinkoje: <https://github.com/NenadZG/mBot-simulation>

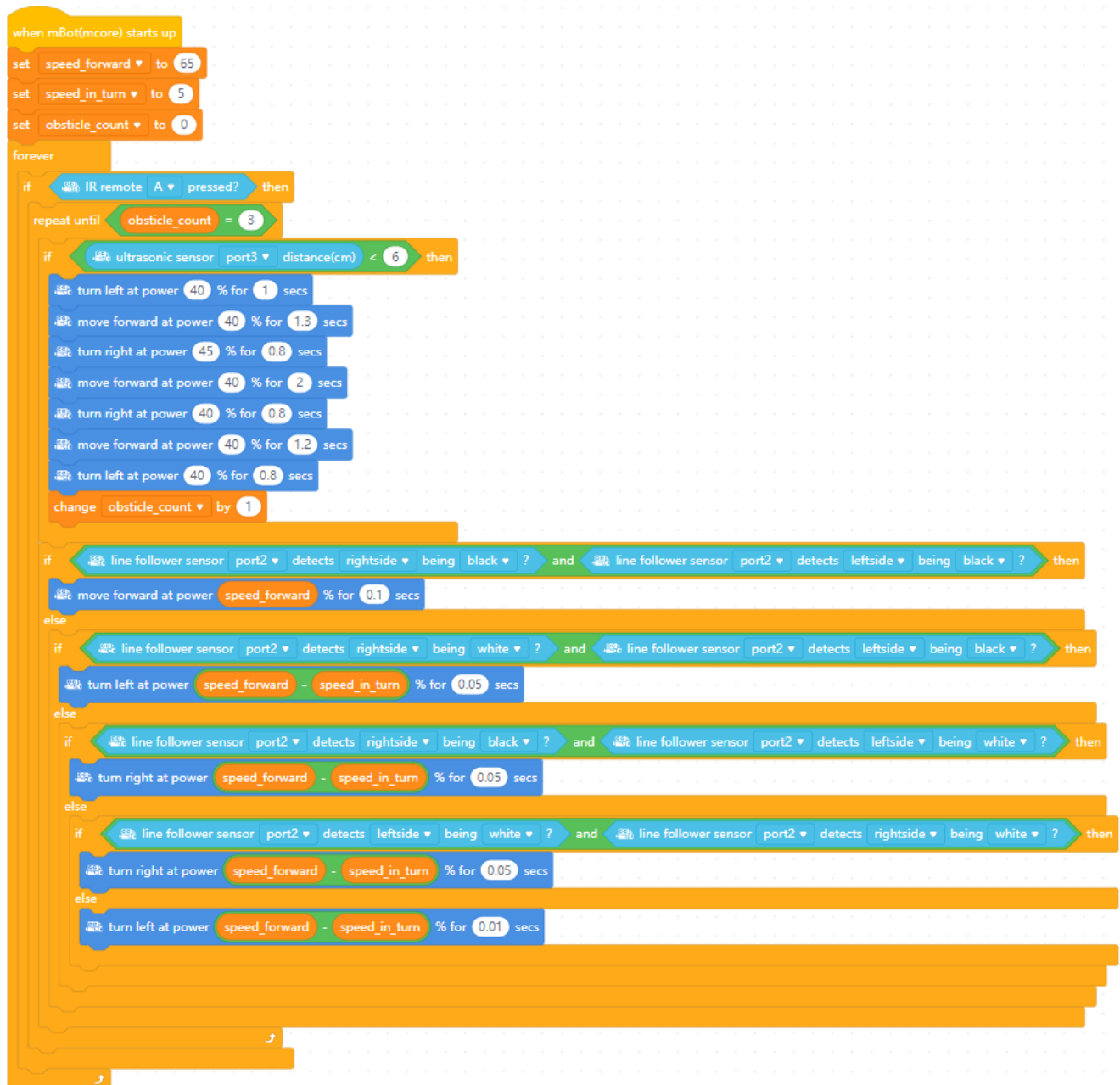
## 2.2. Uždaros trajektorijos apvažiavimas ir kliūtis išvengimas (Veikimas parodytas)

### 2.2.1. Užduoties aprašymas

Robotas paleidžiamas važiuoti. Robotas seka liniją ir važiuoja nubraižyta trajektorija. Ant važiuojamos linijos trajektorijos yra pastatoma kliūtis. Robotas turi tą kliūtį apvažiuoti ir vėl grįžti į trajektoriją ir važiuoti toliau. Robotas apvažiuoja trajektoriją 3 kartus. Ketvirtą kartą aptikęs kliūtį, ją apvažiuoja ir sustoja.

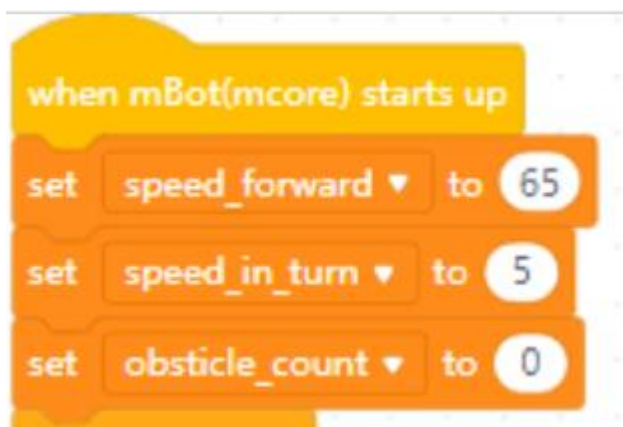
### 2.2.2. Užduoties atlikimas

Užduotis buvo atliekama, kad paspaudus A mygtuką ant pultelio kad robotas pradėta sekti juoda liniją



Pav. 1 Uždaros trajektorijos apvažiavimo algoritmas

Pagal kodą pirmiausia yra parenkami kintamieji kurie bus naudojami roboto algoritme (speed\_forward, speed\_in\_turn, obstacle\_count).



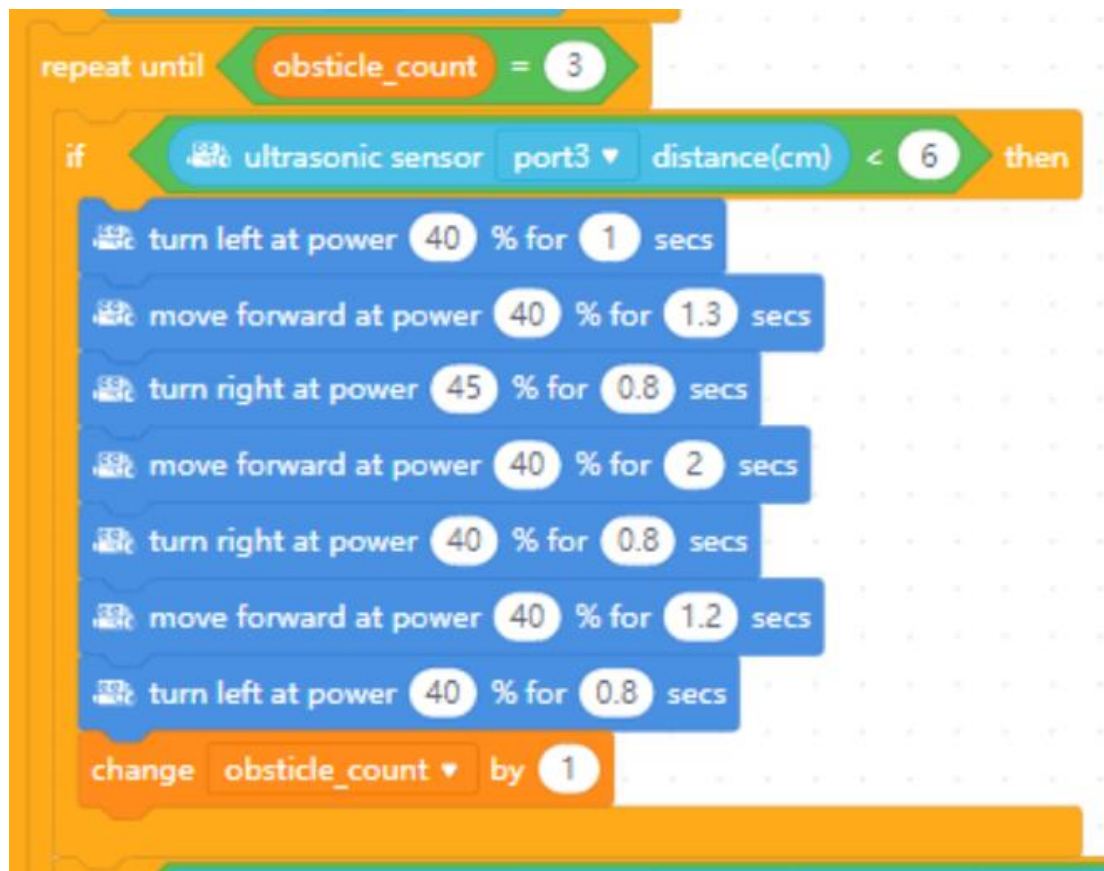
pav. 2 Argumentų parinkimas

Paspaudus A mygtuką robotas pradeda važiuoti tiesiai arba sukasi, priklausomai ar linijos sensorius mato tik juodą liniją pagal kuria važiuoja tik tiesiai, arba jeigu viena iš sensorių pusė mato baltą liniją, tuomet robotas suka į priešingą pusę.



pav. 3 Linijos sekimo algoritmas

Taipogi jeigu roboto garso sensoriai aptinka priekyje kliūtį, tuomet ją bando apvažiuoti atliekant statų posūkį, apvažiavęs kliūtį užskaito jog baigiasi ratas. Robotas kartoją tokius ratus dar 2 papildomus kartus, tuomet sustoja po paskutinės kliūties apvažiavimo.



pav. 4 Kliūtės išvengimo algoritmas

### 2.2.3. Dalinė išvada

Užduočiai atlikti prireikė daug laiko ir net poros paskaitų, padaryti linijos sekimą nebuvo sudėtinga, tačiau duotoje trasoje papildomos kliūtys privertė keisti kodo logiką į labiau universalią. Dėl staus kampo kliūtis teko keisti, kad robotas nebevažiuotu atgal, tačiau važiuotu visuomet į priekį, tik pradėtu sukti taipogi į vieną pusę kol randa vėl kelią. Taipogi baltų linijų kliūtis privertė atitaikyti tinkamą greitį, kad robotas spėtu pravažiuoti pro balta liniją ir neišvažiuotu iš trasos. Su paskutiniu kliūtis išvengimu buvo sudėtinga nustatyti tinkamą greitį posūkiams, kadangi ratai nuolatos skirtingai pasisukdavo, dėl trasos nelygumų arba dėl baterijos problemų.

## 2.3. Parkavimo Sistema (Veikimas neparodytas)

### 2.3.1. Užduoties aprašymas

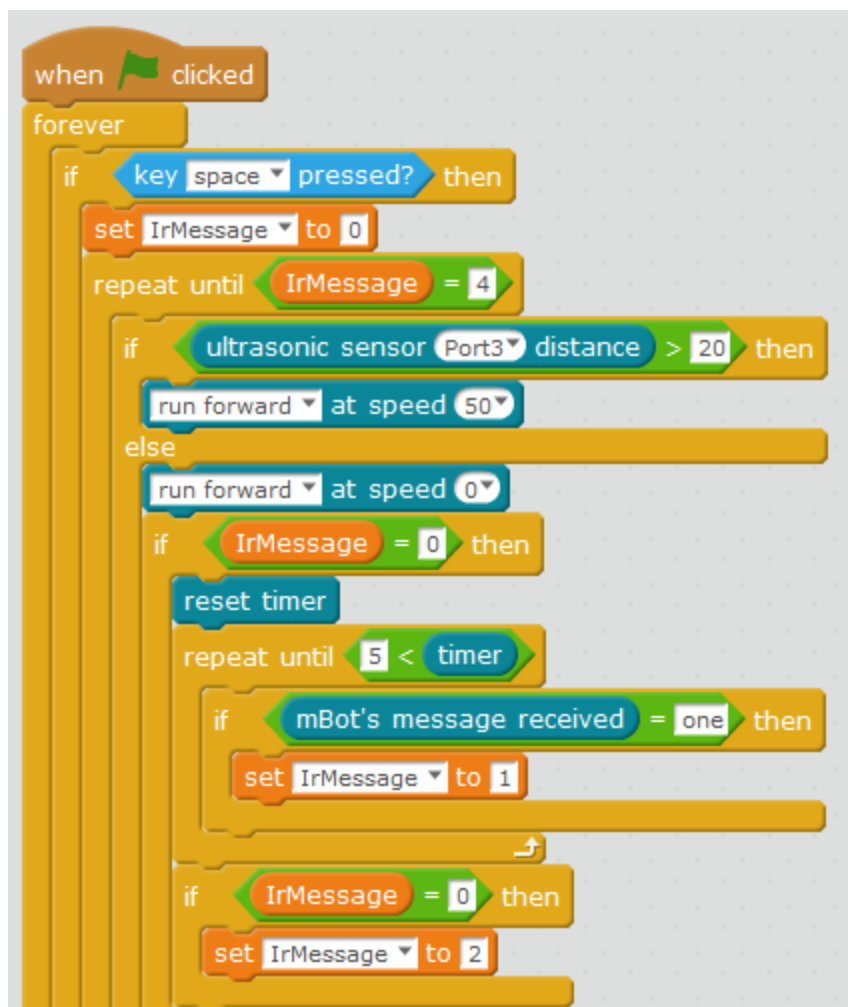
Robotai važiuoja vienas paskui kitą. Pirmas robotas privažiavęs parkavimo vietą (ji apibūdinama kaip kliūtis ). Pirmiausia patikrina ar laisva vieta, t.y sustoje, laukia nuo 3 iki 5 į sek tikrindamas ar gaunamas IR pranešimas. Jei per tą laiką negautas pranešimas, vadinasi vieta laisva ir galima parkuotis. Robotukas pavažiuoja į priekį, apsisuka ir sustoja siunčia IR signalą(pranešimą). **Signalo IR siuntimas ir priėmimas yra kartojamas kas xxxx laiko tarpus** (ciklas).

Kitas robotas privažiavęs kliūtį, sustoja laukia nuo 3 iki 5 į sek tikrindamas ar gaunamas IR pranešimas. Jei gautas pranešimas , robotukas pasisuka į šoną , pavažiuoja ir parkuojasi šalia.

### 2.3.2. Užduoties atlikimas

Roboto veikimas naudojant šitą kodą yra parodytas Vaizdo įrašė: <https://youtu.be/JDTDXpBXqC4>

Parkavimo sistema atlikta naudojant CoppeliaSim robotų simuliacijos aplinką ir dvi senesnes mBlock versijas.



pav. 5 Kliūties privažiavimas ir pranešimo gavimas

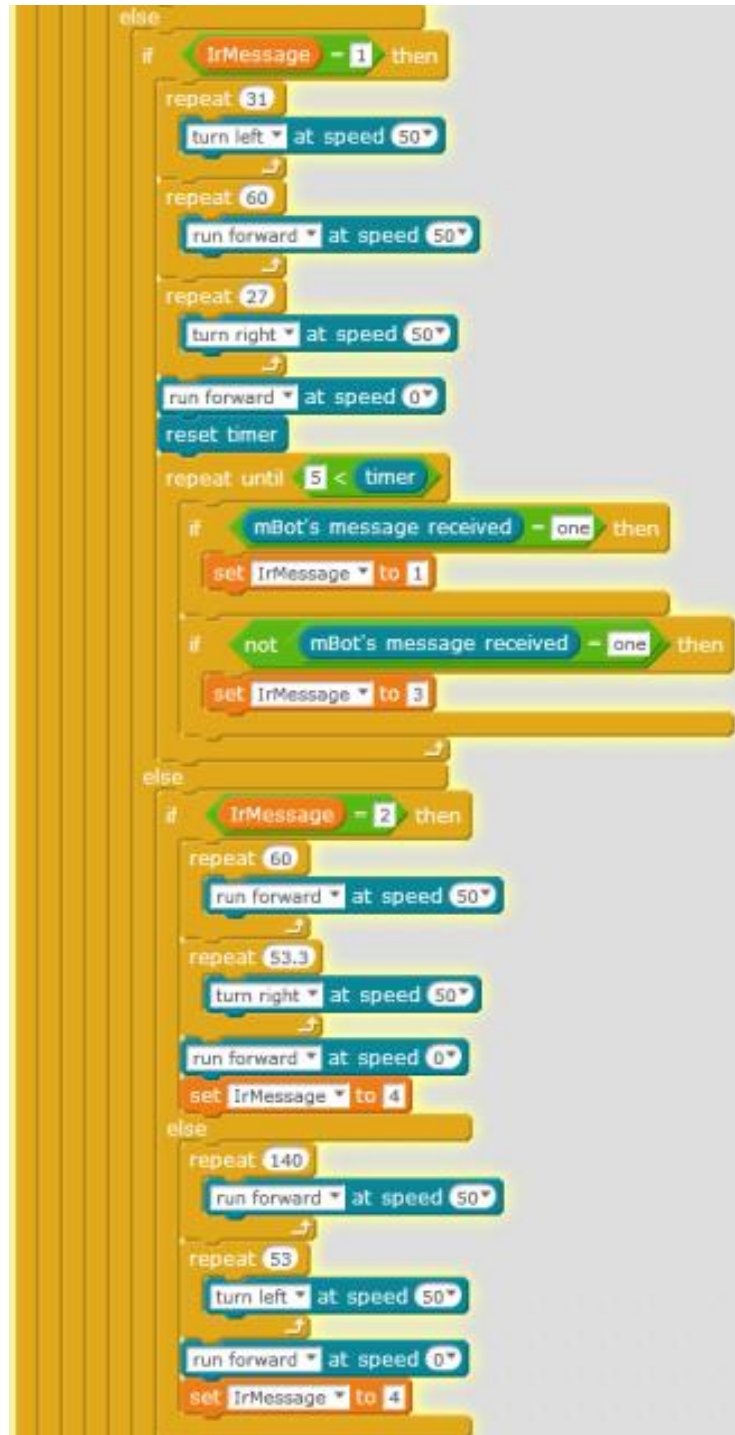
Pirmame kodo fragmente matome aprašytą kintamąjį IrMessage kuris yra reikalingas nustatyti roboto dabartinę būseną.

Algoritme yra naudojamas penkios roboto būsenos:

0. Robotas važiuoja iki kliūtis
1. Robotas gauna pranešimą, kad vieta užimta
2. Robotas negauna pranešimo ir gali parkuotis
3. Robotas negauna pranešimo norint šalia parkuotis
4. Robotas siunčia žinutes kitam robotui jeigu pamato ji su sensorių

Aprašę būsenas galime toliau testuoti kodo aiškinimą.

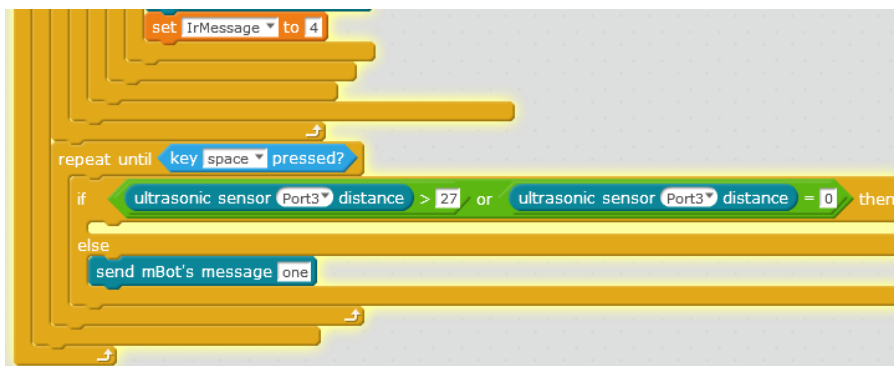
Robotas važiuoja tol kol pamato kliūtį, pamatęs kliūtį robotas sustoja ir penkias sekundes bando gauti pranešimą. Pagal tai ar gavo ar negavo pranešimo yra nustatoma būsena pagal kurią toliau veiks algoritmas.



pav. 6 Pranešimo laukimas ir būsenų pakeitimas



Jeigu robotas negavo pranešimo tada pradeda važiuoti į priekį ir tada pasisuka 180 laipsniu kampu. Pasisukusio roboto būsena nustatoma į ketvirtąją paskutinę būseną kuri skirta pranešimų siuntimui. Robotui gavus pranešimą būsena yra nustatoma į pirmąją būseną. Pakeitus būseną į pirmąją robotas pasisuka į kairę pavažiuoja į priekį atsisuka atgal ir vėl bando parkuotis. Laukia penkias sekundes pranešimo jeigu jo negauna būsena yra nustatoma į trečią. Atnaujinus būseną į trečią robotas pavažiuoja į priekį šalia kito roboto ir apsisuka 180 laipsnių kampu. Pasukus robotą būsena yra nustatoma į ketvirtą kuri yra paskutinė.



*Pav. 7 pranešimų siuntimas*

Paskutinės būsenos metu robotas siunčia pranešimus kitiems robotams kurie yra arti to roboto kuris siunčia.

### 2.3.3. Dalinė išvada

Algoritmas buvo parašytas nesudėtingai, tačiau kadangi robotas retkarčiais važiuodavo skirtingai nes procesoriaus naudojimo lygis skiriasi reikėjo labai tiksliai atitaikyti robotų greičius. Naudojantis rasta mBot simuliacija reikėjo būti išradingiems. Reikėjo pakoreguoti turimą mBot roboto kodą pranešimų siuntimo funkcionalumui pridėti. Buvo problemų paleidžiant du robotus vienu metu. Sugalvojau paleisti robotus per skirtingas mBlock versijas kurios yra skirtingose vietose. Susitvarkius robotą ir atitaikius visus algoritmo parametrus abu robotai parkuojasi be problemų.

## 2.4. Estafetes pradavimas (Veikimas neparodytas)

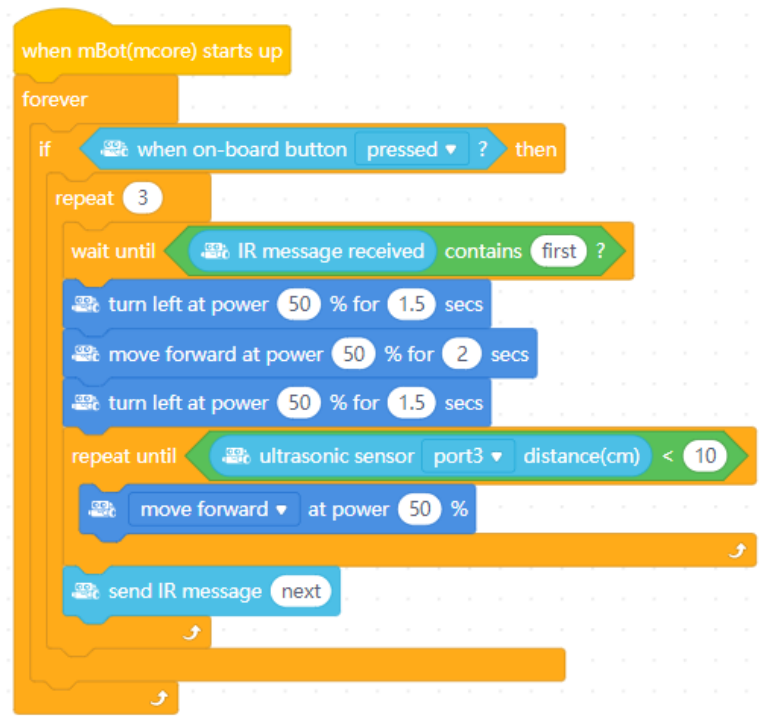
### 2.4.1. Užduoties aprašymas

Ijungti robotai stovi vienas priešais kitą atstumu yyyyy. Robotų įjungimas atliekamas mygtuku **OnBord**. Vienas robotas pradeda važiuoti. Robotui priartėjus prie kito roboto per atstumą .xxxx., jis sustoja ir perduoda IR pranešimą. Kitas robotas gavęs pranešimą apsisuka ir važiuoja tiesiai atstumą kkkkk. Po to apsisuka ir vėl grįžta į pradinę padėtį. Sustoja ir perduoda pranešimą kitam robotui. Pirmas robotas priėmęs šį pranešimą apsisuka, ir važiuoja tiesiai atstumą xxxx. Po to apsisuka ir vėl grįžta į pradinę padėtį. Taip pakartojamas ciklas 3 kartus. Baigus ciklą abu robotai sustoja.

### 2.4.2. Užduoties atlikimas



Pav. 8 Estafetės perdavimo pirmojo roboto algoritmas



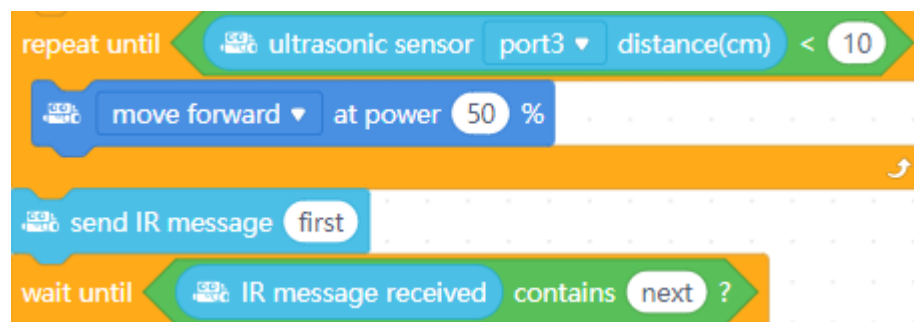
pav. 9 Estafetės perdavimo antrojo roboto algoritmas

Abiem robotam yra sukuriami du algoritmai, kadangi jų veikimas nors ir panašumų turi, tačiau vistiek veikia skirtingai. Abu robotai yra įjungiami naudojant ant roboto esančiais mygtukais „onboard“.



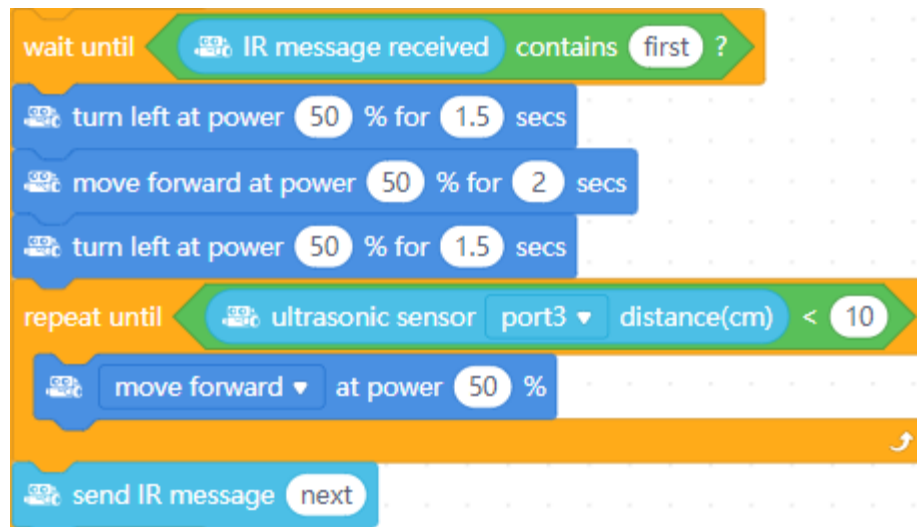
pav. 10 onBoard roboto įjungimas

Pirmasis robotas juda į priekį, o antrasis laukia kol jam atsiunčiama komanda „first“. Pirmasis robotas važiuoja į priekį kol pasiekia antrąjį robotą ir išsiunčia jam pranešimą „first“ ir tuomet sustoją laukdamas atsako „next“ iš antrojo roboto.



pav. 11 Pirmojo roboto važiavimo algoritmas iki antrojo roboto ir žinutės išsiuntimas

Tuo tarpu antrasis robotas gavęs žinutę „first“ apsisuka ir pavažiuoja atgal, vėl apsisuka ir grįžta atgal kol pasiekia pirmąjį robotą, tuomet irgi siunčia žinutę „next“.



pav. 12 Antrojo roboto važiavimo algoritmas gavus žinutę ir perdavimas žinutę atgal pirmajam

### 2.4.3. Dalinė išvada

Algoritmo visos dalys aprašytos ir atrodo iš pirmo žvilgsnio, jog turētu veikti, tačiau kodo nespėjome ištestuoti per paskaitą ir dėl to nežinome ar nustatyti greičiai yra tinkami ir ar robotai tuomet nenuvažiuotu į šoną.

### 3. Išvados

Užduočių visų neatlikome, dėl to nes prie pirmosios užduoties įgyvendinimo užtrukome per ilgai. Darant pirmąją užduotį, nors ir algoritmas buvo padarytas greitai, tačiau daug laiko užtrukome, kol nustatėme tinkamus greičius posūkiams ir pavažiavimam, Taipogi papildomos kliūtims reikėjo apgalvoti kaip algoritmą galima pakoreguoti, tačiau galų gale robotas atliko reikalaujamą užduotį. Didžiausia kliūtis rašant algoritmus buvo laiko stoka per paskaitą, kadangi užstrigus prie vieno algoritmo kitiems algoritmam laiko nebėra. Pramokome naudotis linijos sekimo sensoriumi ir pakoregavom duotąjį sekimo kodą kad sektu tiksliau ir neišklystu iš trasos labai lengvai, taipogi išbandėme žinučių siuntimą, nors ir nespėjome atsiskaityti užduočių su žinutėmis.