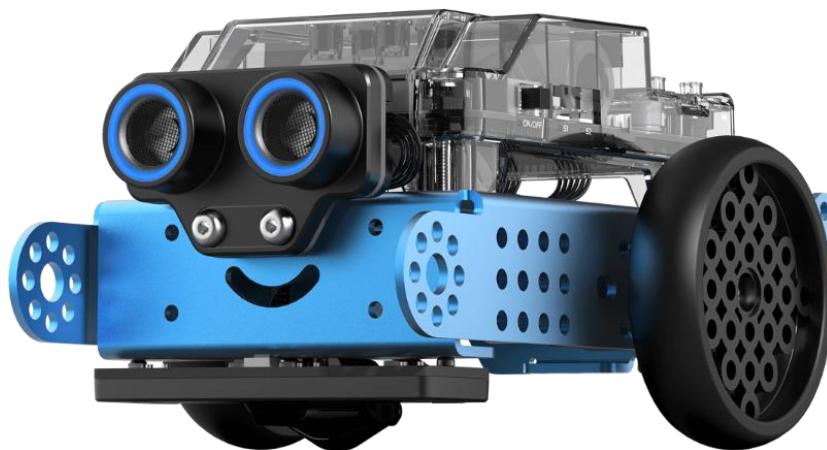


7  
aktivitātes



# Makeblock mBot2 mācību aktivitātes

---

Šajā grāmatā ir apkopota ražotāju un Insplay Latvija un Igaunijas programmas un konkursu dalībnieku aktivitāšu idejas, lai palīdzētu vieglāk uzsākt darbu ar Makeblock mBot2. Grāmatā ir aprakstīti pirmie soļi pēc mBot2 konstruktora salikšanas.

Mēs ceram, ka jūs no šejienes smelsities jaunu iedvesmu un aicinām dalīties savos mācību stāstos ar citiem, izmantojot [Robotikas bērnodārza Facebook grupu](#).

*Pēdējā versija: 02.01.2023.*

# Saturs

1. Kustēties! .....	3
2. Sensori = dati .....	7
3. Klausies mBot2 .....	13
4. Redzēt caur skaņu .....	16
5. Ievērojamu vietu apskate .....	19
6. Pavasara dabas atklāšana .....	26

# 1. Kustēties!

**Autors:** Makeblock

**Sarežģītums:** Iesācēji

**Aktivitātes ilgums:** 45 minūtes

**Nepieciešamie resursi:** mBot2, dators (ar USB portu), USB-C vads, mBlock programmatūra, A3 papīrs, pildspalvas

## Mērķis

- Iemācīties precīzi braukt ar mBot2.
- Izprast dažādus programmēšanas blokus, ko var lietot mBot2 vadīšanai.

## Aktivitātes apraksts

*Iesildīšanās (5 minūtes)*

Roboti ir sastopami dažādās vietās, tie var būt mājās, darba vietās vai citur. Tiem ir dažādas darbības iespējas un ir radīti specifiskiem mērķiem. Uzdodiet skolēniem, lai tie nosauc trīs robotus ar kuriem tie sastopas ikdienā. Protams, ir vairāk robotu veidu, nekā jūs variet izdomāt, tāpēc sameklējiet internetā vēl trīs robotus, kuri ir sastopami cilvēku ikdienā.

Pēc tam izrunājiet par profesijām, kuras ikdienā strādā ar robotiem, par cik nākotnē darbs ar robotiem paliks ar vien biežāk sastopams un visticamāk daudziem skolēniem, būs ikdienā jāstrādā ar tiem. Palūdziet bērniem nosaukt vismaz trīs profesijas vai biznesus, kuros ir ļoti svarīgi, lai roboti strādātu precīzi.

Kad esiet to izrunājuši, iepazīstiniet ar mBot2 – programmējamu robotu, kurš ir aprīkots ar dažādām detaļām, kas ļauj tam sajukt, rīkoties un komunicēt ar apkārtējo vidi. mBot2 ir paredzēts, lai skolēni varētu iemācīties datorzinības un tehnoloģijas tādā veidā, kas ir pēc iespējas vairāk pietuvināts reālās dzīves pielietojumiem. To var programmēt izmantojot mBlock, kur programmēšana var būt tik viegla kā bloku ievilkšana un salikšana nepieciešamajā secībā. Tajā ir iebūvēti dažādi sensori, bet lielāko daļa tiks apskatīta tālākajās aktivitātēs.

*Pirmo darbību apgūšana (10 minūtes)*

Kad jūs sāksiet programmēt mBot2, jūs pamanīsiet, ka programmēšanas bloki ir ļoti dažādi un to iespējas ir plašas. Jūs variet likt tam kustēties, nepieciešamos blokus jūs atradīsiet mBlock 5 kategorijā „Chasis”. Šie programmēšanas bloki ir zili.

Zemāk ir redzami daži piemēri blokiem. Daži no tiem izskatās līdzīgi, tomēr katram no tiem ir sava atšķirīga funkcionalitāte. Šai nodarbībai izmantojiet „Live mode”.



moves forward ▾ at 50 RPM for 1 secs

Šis programmēšanas bloks ļauj jums likt mBot2 dodies uz priekšu, atpakaļ, pa labi vai pa kreisi noteiktā ātrumā. Tāpat tas ļauj noteikt darbības ilgumu sekundēs.



Kamēr šis piemērs parāda to, kā jūs variet likt 1 sekundi kustēties mBot2 uz priekšu ar ātrumu 50 rotācijas minūtē. Šis ir noderīgs, kad mBot2 ir nepieciešams virzīt kravu uz priekšu.



Šis programmas bloks liek robotam apstāties. Šis var būt noderīgs bloks, testēšanas fāzē. Ja programma nestrādā, tā kā jūs vēlaties, tad ar šo jūs variet uzreiz tam likt apstāties. Piemēram, jūs variet likt robotam apstāties, kad tiek nospiests A poga uz CyberPi. Lai to izdarītu, izmantojiet sekojošos blokus:








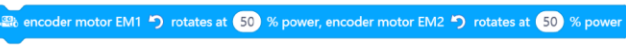
Šis programmas bloks liek mBot pagriezties pa noteiktiem grādiem. Jūs variet izvēlēties vai rotācija ir pa labi vai pa kreisi. Piemēram, kad programma ir iestādītā kā zemāk redzams, mBot2 var tik kontrolēts izmantojot bultu pogas. Kad tiek nospiesta labā bultas poga, mBot2 pagriezīsies 90 grādus pa labi, kamēr nospiežot kreiso bultas pogu, 90 grādi pa kreisi.



Šis programmas bloks liek mBot2 kustēties uz priekšu vai atpakaļ noteiktu distanci. Piemēram, zemāk redzamajā piemērā mBot2 kustēsies 100cm uz priekšu. Lai uzsāktu kustību, pārvirziet CyberPi kursoru uz priekšu.

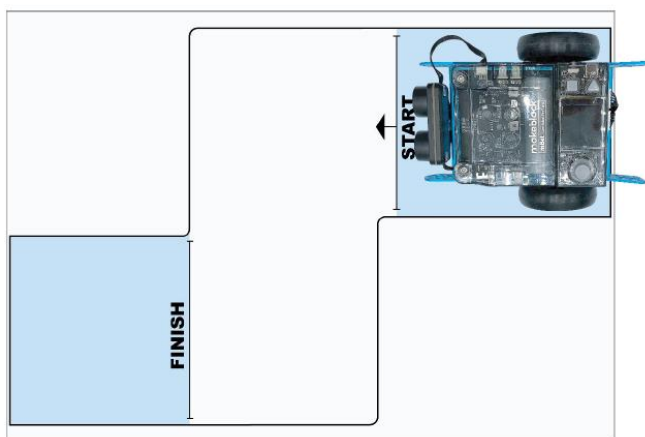


Pēc pamatsoļu apskatīšanas, nākamais solis ir izmēģināt dažus no blokiem. Zemāk redzami daži bloku pāri. Izmēģiniet katru blakus esošo bloku pāri. Kas notiek? Kādas ir atšķirības starp šiem blokiem?

	Option 1	Option 2
1		
2		
3		

### Izmēģināšana (25 minūtes)

Tagad ir laiks likt mBot2 strādāt. Paņemiet A3 lapu un uzzīmējiet labirintu, kuram mBot2 būs jāizbrauc cauri. Netaisiet to pārāk sarežģītu un ņemiet vērā robota parametrus. mBot2 nav jāatrod pašam savs ceļš, jums ir nepieciešamas robotam pateikt priekšā, kas ir jādara. mBot brauks to ceļu, ko būsiet ieprogrammējuši. Nepieciešama iedvesma? Te būs piemērs:



Izmantojiet iepriekš pārrunātas funkcijas un blokus, tomēr atcerieties, ka variet arī eksperimentēt. Kad veiciet šo aktivitāti ir labi izmantot sekojoši soļu plānu:

#### 1. solis – Ko jūs gribiet darīt?

Kādu maršrutu jūs gribiet braukt?

No kādām daļām sastāv labirints?

Cik garas distances robotam ir nepieciešams veikt?

Vai robotam ir nepieciešams pagriezties? Uz kuru pusi un pa cik grādiem?

#### 2. solis – Kas jums ir nepieciešams?

Vai ir kaut kas papildus nepieciešams mBot2, lai veiktu uzdevumu?

#### 3. solis – Kādus programmēšanas blokus jums ir jāizmanto, lai mBot2 brauktu?

Kā jūs liksiet mBot2 braukt?

Kādus blokus jūs izmantosiet?

#### 4. solis – Testēšana un ieviešana

Vai pirmā versija ir gatava? Pārbaudiet to! Pierakstiet, ko varētu uzlabot.

Strādājiet pie uzlabojumiem, līdz mBot2 tiek cauri bez nekādām problēmām.

Kad tas ir paveikts un robots tiek cauri trasei bez problēmām, pamēģiniet grūtāku trasi.

*Noslēgums (5 minūtes)*

Vai jums sanāca likt robotam izbraukt cauri labirintam bez nekādām problēmām? Nodarbības beigās pārrunājiet šādus jautājumus:

- Kas sanāca labi?
- Kas varētu būt labāk?
- Kura nodarbības daļa bija visvieglākā un kura bija visgrūtākā?
- Par kurām lietām vēl būtu nepieciešams vairāk izrunāt/apskatīt?

## 2. Sensori = dati

**Autors:** Makeblock

**Sarežģītums:** Iesācēji

**Aktivitātes ilgums:** 45 minūtes

**Nepieciešamie resursi:** mBot2, dators (ar USB portu), USB-C vads, mBlock programmatūra

### Mērķis

- Iepazīties ar mBot2 sensoriem un saprast to darbību.
- Saprast atšķirību starp „live“ un „upload mode“.
- Kā dati no sensoriem parādās CyberPi.

### Aktivitātes apraksts

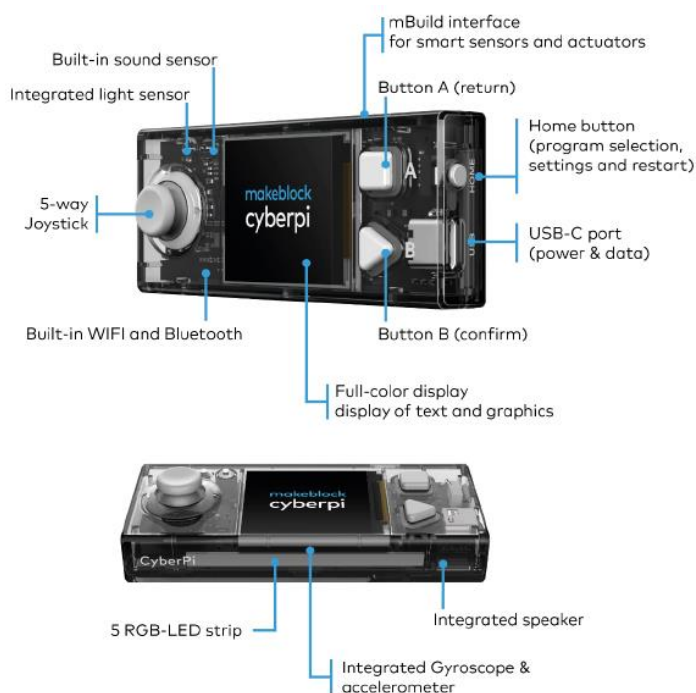
*Iesildīšanās (5 minūtes)*

Datu vācoši sensori ir sastopami dažādās mums ikdienā redzamās vietās. Daudz vairākās nekā sākotnēji varētu likties. Piemēram, lapas, kuras iedegās balstoties uz to vai ārā ir gaišs vai tumšs. Mēģiniet ar skolēniem nosaukt vēl citus piemērus.



mBot augšā ir redzams CyberPi – mazs, programmējams mikrodators. CyberPi saņem komandas caur mBlock programmatūru jūsu datorā un pēc tam nodod šīs programmas mBot2. CyberPi var tikt atdalīts no mBot2.

CyberPi ir daudz dažādas funkcionalitātes, piemēram, mikrofons, skaļrunis un kursoris. Tāpat ir daudzi sensori. Zemāk attēlā vizuāls attēlojums:

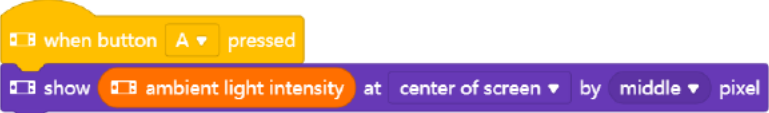
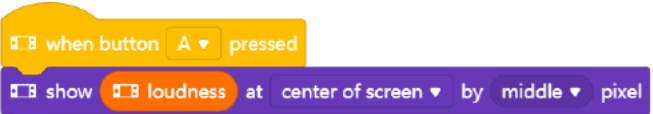


*Pirmo darbību apgūšana (10 minūtes)*

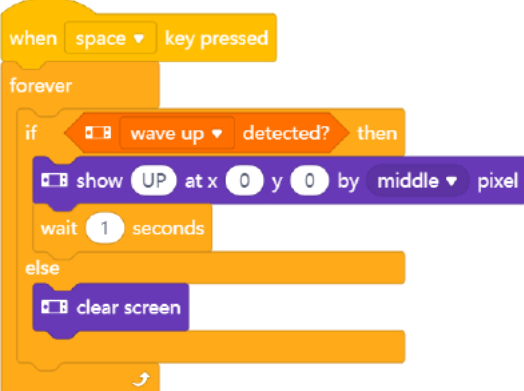
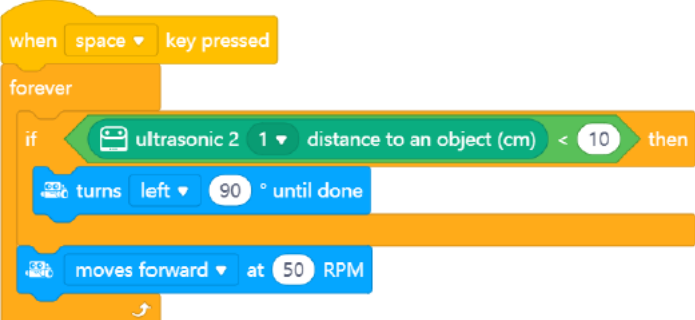
Sākumā iepazīstieties ar katru no sensoriem. Sensori var tik salīdzināti ar cilvēku maņām (garšu, tausti, smaržu, skaņu, redzi). Caur šiem sensoriem mBot2 „redz” tā vidi. Ir vairāki sensori, kas var likt „redzēt”, piemēram:

- Gaismas sensors
- Skaņas sensors
- Žiroskops un akselerometrs
- RGB sensors
- Taimeris

Zemāk ir redzams sensoru saraksts, kā ar programmēšanas piemēri.

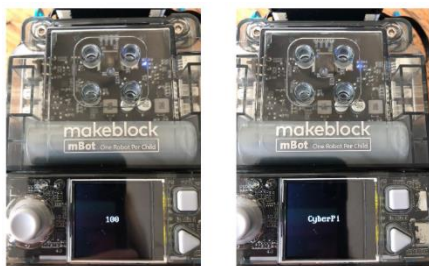
Sensora veids	Ko šis sensors dara?
Gaismas sensors	<p>Gaismai ir noteikts spēks (gaismas enerģija). Gaismas sensors ir ierīce, kas kovertē gaismas enerģiju elektriskajā enerģijā.</p> <p>Piemēram gaismas sensoru lietošanai:</p> <ul style="list-style-type: none"> <li>• Automātiski pielāgotu telefona ekrāna spilgtumu, vadoties pēc vides.</li> <li>• Kontrolētu mājas apgaismojumu vai lai automātiski ieslēgtu gaismas mašīnā, kad tas ir nepieciešams.</li> </ul>
	<p><b>Programmēšanas piemērs</b></p> <p>Programmēšanas piemērā, vides gaismas enerģija tiek parādīta uz CyperPi ekrāna. Jūs variet paplašināt programmēšanu ar likšanu mBot2 apstāties, piemēram, kad vides gaismas enerģija sasniedz konkrētu vērtību, piemēram, paliek tumšs.</p> 
Sensora veids	Ko šis sensors dara?
Skaņas sensors	<p>Skaņa ir mehāniska vibrācija, kas parādās kā viļņi. Gaisā tās ir spiediena un biežuma izmaiņas. Ja vibrācija ir dzirdamā spektrā (starp 16 un 20000 vibrācijām/sekundē) un pietiekošā intensitātē, mēs to varam dzirdēt kā toni vai skaņu.</p> <p>Skaņas spēks tiek saukts par skaņas intensitāti.</p> <p>Bieži skaņas sensors tiek lietots diviem mērķiem:</p> <ul style="list-style-type: none"> <li>• Telefona zvaniem, skaņas ierakstiem.</li> <li>• Ar balsi kontrolētām sistēmām, piemēram, Apple Siri vai gudrajās mājās esošajām.</li> </ul>
	<p><b>Programmēšanas piemērs</b></p> <p>Programmēšanas piemērā, vides skaņas intensitāte ir attēlota CyperPi ekrānā. Jūs variet paplašināt programmēšanu ar likšanu mBot2 ierakstīti klases skaņas, braucot apkārt, un tad parādot rezultātus.</p> 
Sensora veids	Ko šis sensors dara?
Žiroskops & akselerometrs	<p>Žiroskops mēra apgāšanās kustību, precīzāk pagriezienu/apgāšanās kustības ātrumu. Akselerometrs mēra ātruma izmaiņas. Šos sensorus var ieviest kā elektroniskā komponenta mikromehāniskus komponentus. Abi sensori sniedz atšķirīgu informāciju par atrašanās vietu telpā.</p> <p>Žiroskopu bieži izmanto, lai:</p>



	<ul style="list-style-type: none"> <li>• Nodrošiniet kuģu stabilitāti atklātā jūrā.</li> <li>• Sabalansējiet Segway vai "hoverboard", lai ar to pārāk ātri neapgāztos.</li> </ul> <p>Paātrinājuma sensors savukārt norāda, vai viedtālrunis ir vērst uz augšu vai uz leju, vai transportlīdzeklis ir nokļuvis avārijā (vardarbīga, ļoti strauja ātruma maiņa), lai pēc tam iedarbinātu drošības spilvenus.</p> <p><b>Programmēšanas piemērs</b></p> <p>Tālāk esošajā programmēšanas piemērā displejā tiek rādīta mBot2 slīpuma kustība. Varat paplašināt programmēšanas piemēru tā, lai mBot2 nepārtraukti mērītu un ierakstītu displejā apgāšanās kustību klasē apļa laikā.</p> 
<b>Sensora veids</b>	<b>Ko šis sensors dara?</b>
Ultraskaņas sensors	<p>Skaņa, kas ir vibrācija blīvuma un spiediena izmaiņu veidā, gaisā izplatās nemainīgā ātrumā viļņu veidā (skaņas ātrums gaisā apm. 334 m/s). Jo augstāka ir vibrācija, jo augstāka ir skaņa - līdz robežai, kuru pārsniedzot, cilvēki nevar atpazīt šo skaņu. Šis ļoti augstās vibrācijas sauc par ultraskaņu. Tā kā skaņas ātrums gaisā ir zināms, skaņu var izmantot, lai noteiktu attālumus līdz objektiem; lai to izdarītu, izstaro skaņu un mēra laiku, līdz šī skaņa tiek atspoguļota atpakaļ no objekta. Šos atstarotos skaņas viļņus sauc arī par atbalsi. Šim nolūkam parasti izmanto ultraskaņas.</p> <p>Piemēram, bieži izmanto ultraskaņas sensoru:</p> <ul style="list-style-type: none"> <li>• Attēla veidošanas procedūrai, piemēram, grūtniecības laikā (katrs pikselis ir attāluma mērījums);</li> <li>• robotu kontrolē, lai novērstu sadursmes.</li> </ul> <p><b>Programmēšanas piemērs</b></p> <p>Programmēšanas piemērā ultraskaņas sensors tiek izmantots, lai novērstu mBot2 iebraukšanu šķērslī. Kad mBot2 atrodas mazāk nekā 10 cm no šķēršļa, robots veiks 90 grādu pagrieziena pa kreisi un tad vienkārši turpinās braukt.</p> 
<b>Sensora veids</b>	<b>Ko šis sensors dara?</b>
RGB sensors	<p>Četrus RGB sensoru veido četri atsevišķi gaismas un krāsu sensori. Tie mēra gaismas intensitāti, kas ietilpst sensorā no gaismas spektra sarkanajām, zaļajām un zilajām zonām. Tas ļauj sensoram noteikt objektu krāsas tieši tā</p>

	<p>priekšā, piemēram, marķējumus uz zemes, kā arī ļauj robotam sekot melnai līnijai, lai orientētos.</p> <p>Četrus RGB sensors tiek plaši izmantoti tādās lietojumprogrammās kā:</p> <ul style="list-style-type: none"> <li>parādīt noliktavas robotam ceļu cauri noliktavai (arī dažādas krāsas dažādiem ceļiem);</li> <li>pārlicinieties, ka krāsošanas laikā tiek izmantota pareizā krāsa.</li> </ul> <p><b>Programmēšanas piemērs</b></p> <p>Programmēšanas piemērā mBot2 virzās uz priekšu, kad robots redz balto krāsu.</p>
<p><b>Sensora veids</b></p>	<p><b>Ko šis sensors dara?</b></p>
<p>Taimeris</p>	<p>Taimeris ir sava veida hronometrs, kas parāda laiku sekundēs kopš CyperPi ieslēgšanas vai atiestatīšanas. Šo skaitītāju var arī iestatīt uz nulli ar komandu, lai atvieglotu laika mērīšanu.</p> <p>Piemēram, varat izmantot taimeri, lai sacenstos vairākos mBotos vienam pret otru un precīzi izmērītu, cik ātri viņi ir finiša taisnē.</p> <p><b>Programmēšanas piemērs</b></p> <p>Programmēšanas piemērs parāda, kā iestatīt un lietot taimeri. Nospiežot taustiņu A, taimeris tiek atiestatīts uz nulli un pēc tam pastāvīgi tiek rādīts displejā. Varat beigt displeju un programmu kopumā, nospiežot B.</p>

Kopā izmēģiniet augstāk minētos piemērus, kā padomājiet arī par papildus funkcijām, ko variet iekļaut. Šim būs nepieciešams izmantot „Live mode“.



Testējot programmas piemērus, jūs novērojāt, ka katrs sensors kaut ko parāda uz CyberPi ekrāna. Piemēram, gaismas sensors parāda gaismas intensitāti. Jūs variet parādīt sensoru datus mBot2 displejā, piemēram, ka ciparus, bet ne tikai. Šos programmas blokus var atrast zem mBlock kategorijas „Display“.

Programmēšanas piemēri	
Līnijas diagramma	
Līnijas diagramma (krāsas)	
Tabula	

Programmā mBlock izveidotā datorprogramma ir jāpārsūta uz CyberPi. CyberPi liek mBot2 darboties saskaņā ar komandām, kuras ierakstījāt mBlock. Rakstot datorprogrammu, varat tieši pārbaudīt to, ko veidojat. Jūs to darāt, izmantojot tiešraides režīmu (live mode). Jūs izvēlaties tiešraides režīmu, pārvietojot Mode slēdzi pa labi. Lietojot tiešraides režīmā, CyberPi nedrīkst atvienot no datora!

CyperPi ir arī augšupielādes režīms. Augšupielādes režīmā programma tiek pārsūtīta uz CyperPi. Programma tiek saglabāta CyperPi, līdz to aizstājat ar citu programmu. Jūs varat vienkārši atvienot CyperPi.

Zemāk esošajā diagrammā varat redzēt atšķirību starp tiešraides režīmu un augšupielādes režīmu.

Režīms	Atšķirības
Tiešraides režīms (Live mode)	<ul style="list-style-type: none"><li>• Programma atrodas jūsu datorā. Tā netiks saglabāta CyperPi.</li><li>• CyperPi ir jābūt pieslēgtam pie jūsu datora.</li><li>• mBlock programmas rediģētājam vienmēr ir jābūt atvērtam uz jūsu datora.</li><li>• Šis režīms tiek lietots fāžu programmēšanā.</li></ul>
Augšupielādes režīms (Upload mode)	<ul style="list-style-type: none"><li>• Programma atrodas un tiek saglabāta uz CyperPi nevis uz jūsu datora.</li><li>• CyperPi nav nepieciešams būt pievienotam pie datora.</li><li>• mBlock programmai nav jābūt atvērtai.</li><li>• Programma saglabājas uz CyperPi līdz tā tiek aizstāda ar citu.</li></ul>

#### *Izmēģināšana (25 minūtes)*

Jūs jau esat daudz iemācījušies par mBot2 sensoru un to, kā parādīt datus no sensoriem CyperPi displejā. Tagad jūs pats ievāksiet datus, izmantojot mBot2 sensorus, un parādīsiet tos CyperPi displejā. Jūs to darīsiet, braucot ar mBot2 pa klasi vai skolu. Tas, kādus datus apkopojat un kā tos rādīt CyperPi displejā, ir atkarīgs no jums.

Izmantojiet iegūtās zināšanas. Protams, jūs varat veikt daudz eksperimentu ar dažādām programmēšanas iespējām mBlock. Uzdodiet līdzīgus jautājumus, kā to darījāt iepriekšējā aktivitātē.

#### *Noslēgums (5 minūtes)*

Kādus datus jūs ieguvāt ar mBot2 palīdzību? Kā jums ar to veicās? Vai jums izdevās redzēt visus datus uz CyperPi ekrāna?

Nodarbības beigās pārrunājiet šādus jautājumus:

- Kas sanāca labi?
- Kas varētu būt labāk?
- Kura nodarbības daļa bija visvieglākā un kura bija visgrūtākā?
- Par kurām lietām vēl būtu nepieciešams vairāk izrunāt/apskatīt?

## 3. Klausies mBot2

**Autors:** Makeblock

**Sarežģītums:** Iesācēji

**Aktivitātes ilgums:** 45 minūtes

**Nepieciešamie resursi:** mBot2, dators (ar USB portu), USB-C vads, mBlock programmatūra

### Mērķis

- Apgūt kā izmantot skaļruņus programmēšanā un robotikā.
- Iemācīties kā ierakstīt un atskaņot atpakaļ skaņas.
- Iemācīties kā paralēli palaist vairākus programmēšanas uzdevumus.

### Aktivitātes apraksts

*Iesildīšanās (5 minūtes)*

No sākuma izrunājiet par dažādām lielajām skaļruņu sistēmām, piemēram, kinoteātros, koncertu zālēs vai pasākumu telpās. Šie skaļruņi ir domāti, lai pastiprinātu skaņu. Kamēr mazie skaļruņi tiek izmantoti dažādās ierīcēs, piemēram, telefonos, televizoros, mūzikas instrumentos. Kopā ar klasi atrodiat vēl dažus piemērus.

Skaļruņa mērķis ir atskaņot skaņu, lai to izdarītu skaļrunis pārvērš elektromagnētiskos viļņus skaņas viļņos. Papildus tam, šis process var arī notiek pretēji. Mikrofonu ir ierīces, kuras pielieto to pašu principu, tikai pārvēršot skaņas viļņus elektromagnētiskajos.

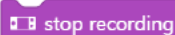
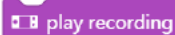
mBot2 ir gan skaļrunis, gan mikrofons. Mikrofons atrodas augšējā kreisajā CyberPi pusē, kamēr skaļrunis priekšējā malā.



Pielietojot mikrofonu un skaļruni kopā ar citiem sensoriem, mBot2 var piemēram, atskaņot konkrētā laikā ierakstu. Piemēram, iedomājieties sabiedrisko transportu, kur autobuss paziņo par nākamo pieturu. Ir vairāki piemēri, kur var abus pielietot, kopā izdomājiet vēl dažus pirms virzāties uz nākamo aktivitātes daļu.

*Pirmo darbību apgūšana (10 minūtes)*

mBlock 5 ir vairāki programmas bloki, kurus jūs variet lietot programmējot skaļruni un mikrofonu. Jūs šos blokus atradīsiet zem „Audio” kategorijas un tie būs violetā krāsā. Zemāk ir daži programmēšanas piemēri:

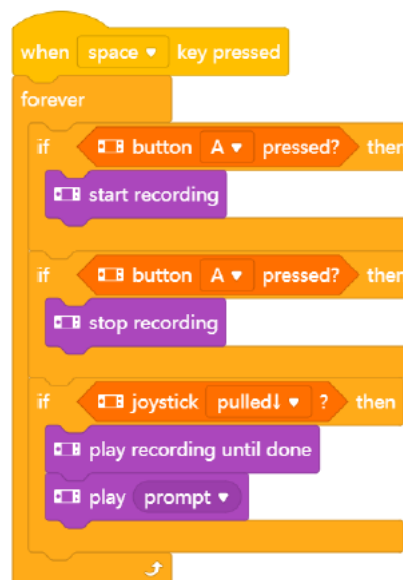
Ierakstot un atskaņojot skaņu, faktiski vienmēr varat izmantot trīs kodu blokus. Izmantojot programmēšanas bloku "start recording", jūs ļaujat CyperPi sākt pārkodēt skaņu. Mikrofona elektriskais signāls tiek saglabāts CyperPi atmiņā, un ierakstīšana automātiski tiek pārtraukta pēc 10 sekundēm. Varat arī norādīt, kad ierakstam jāpārtrauc. Tas tiek darīts ar programmēšanas bloku "stop recording". Ierakstītā skaņa tiek īslaicīgi saglabāta CyperPi atmiņā. Vai jūs izslēdzat mBot? Tad ierakstītā skaņa tiek zaudēta. Skaņas atmaksa tiek kontrolēta ar programmēšanas bloku "play recording".

Programmēšanas piemērā, nospiežot pogu A, CyperPi sāk ierakstīšanu. Pēc 3 sekundēm ierakstīšana tiek pārtraukta. Nospiežot pogu B, jūs dzirdat, kas tika ierakstīts. Izmantojot divus notikumu blokus, divi programmēšanas uzdevumi var darboties neatkarīgi no katra cita, bez nepieciešamības pēc galvenās cilpas.




Jūs tikko iemācījāties atskaņot ierakstītu skaņu. Izmantojot šo programmēšanas bloku, jūs pārliecināsities, ka mikrofona ierakstītā skaņa tiek atskaņota līdz galam. Tas nozīmē, ka turpmāko bloku izpilde tiek bloķēta, līdz šis ir pabeigts. Kad skaņa ir pabeigta, varat likt mBot2 darīt kaut ko citu. Piemēram, izveidojiet noteiktu skaņu, lai zinātu, ka ieraksts ir pabeigts.

Programmēšanas piemērā CyberPi sāk ierakstīšanu, kad nospiežat pogu A. Nospiežot pogu B, ierakstīšana tiek pārtraukta. Ja virzīsiet kursorsvīru uz leju, jūs dzirdēsiet, ko mBot2 ierakstīja. Kad viss ieraksts ir beidzies, jūs dzirdēsiet "uzvednes" skaņu. Programmēšanas bloks šajā piemērā ir ievietots tā sauktajā cilpas struktūrā. Tas nodrošina, ka ievade no pogas un kursorsvīras tiek nepārtraukti uzraudzīta.



Lūdzu, izmēģiniet augstāk minētos piemērus. Padomājiet arī par vismaz vienu veidu, pa paplašināt redzamās programmas un iztestējiet to. Lietojiet „live mode“ testa laikā.

### Izmēģināšana (25 minūtes)

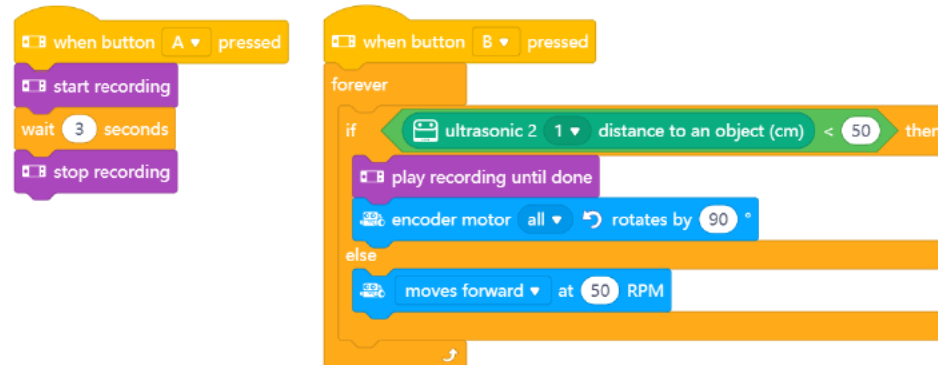
Studenti jau ir daudz uzzinājuši par mBot2 skaļruņiem un mikrofoniem. Tagad viņiem vajadzētu strādāt pie tiem atsevišķi. Viņi arī izmantos ultraskaņas sensoru. Šis sensors var uztvert objektu no attāluma.

Šajā uzdevumā viņi liks mBot2 braukt nejausā aplī pa klasi. Nejaušs nozīmē, ka jūs viņi neveic maršrutu. Ja mBot2 tuvojas šķērslim, piemēram, krēslam, galdam vai mugursomai, viņi dzirdēs skaņu un mBot apgriezīsies. Viņi paši ierakstīs skaņu ar mikrofonu. Šāda skaņa varētu būt, piemēram, skaļa pīkstiena skaņa vai skaņa, kad automašīna palēninās. Varat arī ierakstīt tekstu vai likt mBot2 atskaņot dziesmu.

Ir diezgan grūti pašiem izdomāt datorprogrammu ar šo visu! Par laimi viņi var saņemt palīdzību. Mēs esam izveidojuši dažus piemērus, no kuriem smelties iedvesmu. Viņi var arī paši paplašināt un pielāgot šīs datorprogrammas.

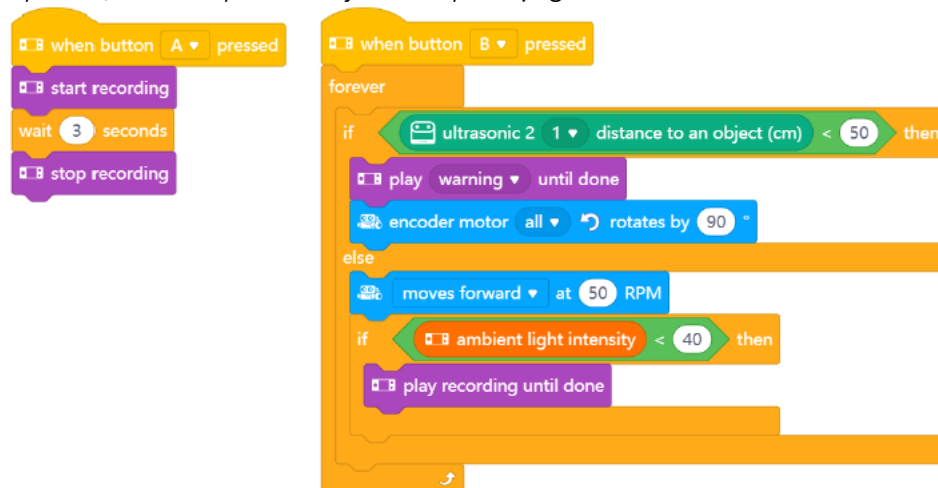
### 1.piemērs

mBot2 atskaņo skaņu, kad tas tuvojas šķērslim un apgriežas otrādi. Skaņa tiek iepriekš ierakstīta.



### 2.piemērs

mBot2 brauc un atskaņo iepriekš ierakstīto skaņu, kad apkārtējā gaisma ir zema. Kad mBot2 tuvojas šķērslim, tas atskaņot brīdinājuma skaņu un pagriežas.



*Noslēgums (5 minūtes)*

Kā uzdevums veicās? Vai mBot2 uzreiz darīja, to ko bijāt ieplānojuši?

Nodarbības beigās pārrunājiet šādus jautājumus:

- Kas sanāca labi?
- Kas varētu būt labāk?
- Kura nodarbības daļa bija visvieglākā un kura bija visgrūtākā?
- Par kurām lietām vēl būtu nepieciešams vairāk izrunāt/apskatīt?

## 4. Redzēt caur skaņu

**Autors:** Makeblock

**Sarežģītums:** Iesācēji

**Aktivitātes ilgums:** 45 minūtes

**Nepieciešamie resursi:** mBot2, dators (ar USB portu), USB-C vads, mBlock programmatūra, mazi šķēršļi (minimālais izmērs 10x10x10 cm), vismaz 1m<sup>2</sup> liela darba virsma.

### Mērķis

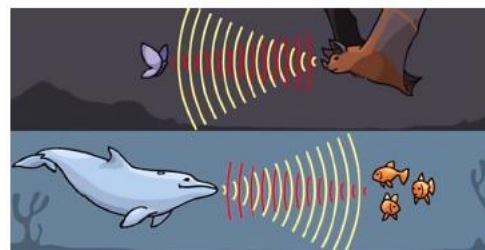
- Šķēršļu noteikšana, izmantojot ultraskaņas sensoru.
- Iemācīties kā likt mBot2 reaģēt uz šķēršļiem un tos apbraukt.

### Aktivitātes apraksts

*Iesildīšanās (5 minūtes)*

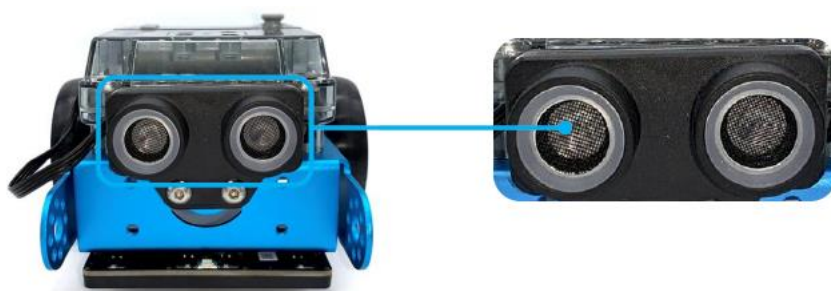
Ultraskaņa ir skaņa ar ļoti augstu frekvenci. Tik augstu, ka cilvēki to nedzird. Ar īpašiem mikrofoniem šo skaņu var ierakstīt un parādīt datorā: šķiet, ka tā ir vibrācija viļņu formā. Tā kā ultraskaņai ir ļoti augsta frekvence, viļņu formas ielejas un virsotnes atrodas ļoti tuvu viena otrai. Ultraskaņa var būt noderīga lieta. Piemēram, jūs varat izmantot ultraskaņas vibrāciju, lai atklātu objektus, vai arī varat to izmantot objektu noteikšanai.

Delfini un abts izmanto ultraskaņu šādā veidā. Tie izstaro ultraskaņas skaņu, piemēram, īsu kliedzienu. Kad šī skaņa atlec no kaut kā, viņi ar ausīm uztver vibrāciju (atbalsi), kas atgriežas. Tā viņi zina, kur atrodas, piemēram, citi dzīvnieki vai kāds šķērslis ir viņu ceļā. Radars wwe šodien izmanto to pašu principu, bet ar radio viļņiem skaņas vietā.



mBot2 ir ultraskaņas raidītājs un uztvērējs. Tie atrodas mBot2 priekšpusē mazajos cilindros, kurus var uzskatīt par "acīm". Šis sensors sūta īsus ultraskaņas skaņu pārrāvumus un klausās, vai nav atbalsi. Vai mBot2 ir tuvu un šķērslis? Pēc tam ultraskaņas skaņa tiek atspoguļota atpakaļ uz mBot2. Pamatojoties uz laiku, kas nepieciešams, lai skaņa atgrieztos sensorā, mBot iekšēji aprēķināja attālumu līdz tam.

Šos datus var izmantot, lai izlemtu, kā robotam jāreaģē: jūs to pasakāt robotam, programmējot. Šāda darbība var būt, piemēram, mBot2 jāpārtrauc braukšana vai jāveic pagrieziena.





*Pirmo darbību apgūšana (10 minūtes)*

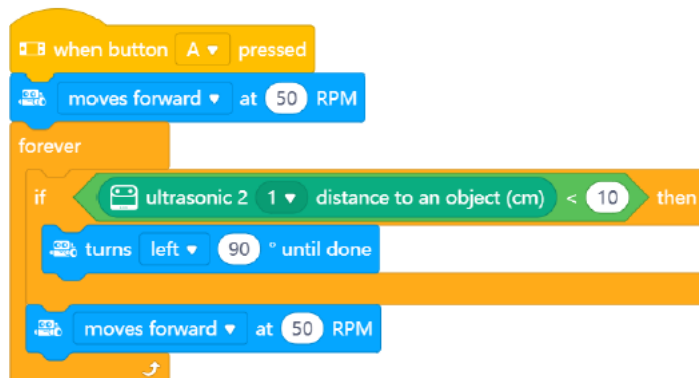

Programmā mBlock ir vairāki ultraskaņas sensora programmēšanas bloki, kurus varat izmantot savās programmās. Lai to izdarītu, no paplašinājuma bibliotēkas jāpievieno paplašinājums "Ultrasonic Sensor 2". Pēc tam koda blokus atradīsiet mBlock bloku lauka kategorijā "Ultraskaņas sensors". Šie programmēšanas bloki ir zaļā krāsā.

Tālāk esošajā tabulā varat redzēt dažus ultraskaņas sensora programmēšanas programmēšanas piemērus.

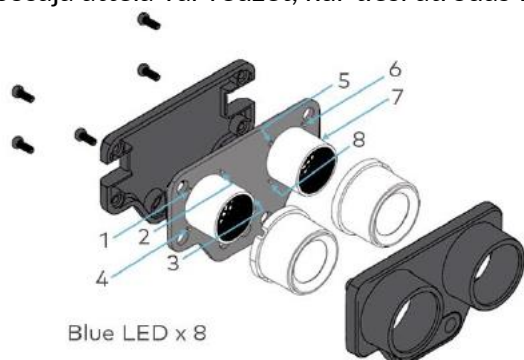
 ultrasonic 2 1 ▼ distance to an object (cm)

Ar šo programmēšanas kodu jūs izmēra attālumu starp sensoru un šķērslī. Sensora diapazons ir no 3 līdz 300 cm. Varat izmantot attāluma vērtību, lai liktu robotam veikt noteiktu darbību. Piemēram, jūs varat novērst mBot2 sadursmi ar šķēršļiem.

Tālāk esošajā programmēšanas piemērā mBot2 virzās uz priekšu. Kad mBot2 atrodas mazāk nekā 10 cm attālumā no šķēršļa, robots pagriežas par 90 grādiem pa kreisi. Pēc tam mBot2 atkal virzās uz priekšu.


 ultrasonic 2 1 ▼ increases ambient light all ▼ brightness by 20 %

Ultraskaņas sensoram mBot2 ir astoņas zilas LED gaismas. Varat izmantot šīs gaismas, lai mBot2 parādītu noteiktas "emocijas", piemēram vai kā vispārēju vizuālu neverbālu saziņu. Ļaujiet LED gaismai spīdēt ļoti spilgti, kad mBot2 ir laimīgs, un mazāk spožu, ja tas ir drūms. Šis ir īpašs programmēšanas bloks, lai parādītu arī iepriekš noteiktas "emocijas". Zemāk esošajā attēlā var redzēt, kur tieši atrodas LED gaismas.



Programmā mBlock ir vairāki programmēšanas bloki, kas ļauj programmēt LED apgaismojuma spilgtumu: jūs varat iestatīt gaismas diožu spilgtumu, samazinot vai palielinot spilgtumu par noteiktu procentu. Vai arī varat iestatīt spilgtumu tieši uz noteiktu procentuālo daļu. Diapazons ir no 0% līdz 100%. To var veikt vai nu vienai LED, vai visiem

vienlaicīgi. Tālāk redzamajā programmēšanas piemērā LED lampas 1 spilgtums ir palielināts par 50%.

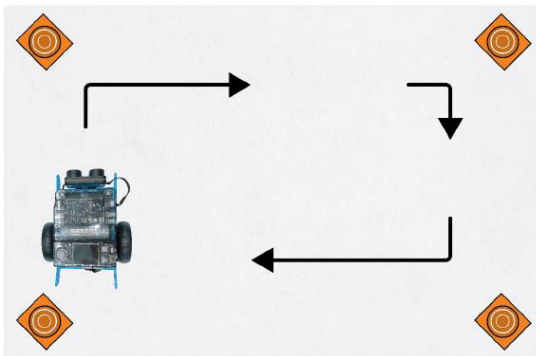
when up arrow key pressed

ultrasonic 2 1 increases ambient light all brightness by 50 %

Lūdzu, izmēģiniet augstāk minētos piemērus. Padomājiet arī par vismaz vienu veidu, pa paplašināt redzamās programmas un iztestējiet to. Lietojiet „live mode“ testa laikā.

### Izmēģināšana (25 minūtes)

Viņi tagad izveidos savu programmu mBlock, izmantojot ultraskaņas sensoru. Šajā uzdevumā viņi gatavojas veikt mBot2 disku bezgalīgos "raundos". To var izdarīt, novietojot objektu katrā virsotnē. Jūs ieprogrammējat mBot2 tā, lai tas apgrieztos katrā objektā un galu galā atgrieztos sākuma punktā. Apskatiet attēlu zemāk.



### Noslēgums (5 minūtes)

Vai sanāca mBot2 izbraukt vismaz vienu apli?

Nodarbības beigās pārrunājiet šādus jautājumus:

- Kas sanāca labi?
- Kas varētu būt labāk?
- Kura nodarbības daļa bija visvieglākā un kura bija visgrūtākā?
- Par kurām lietām vēl būtu nepieciešams vairāk izrunāt/apskatīt?

## 5. Ievērojamo vietu apskate

**Autors:** Makeblock

**Sarežģītums:** Iesācēji

**Aktivitātes ilgums:** 45 minūtes

**Nepieciešamie resursi:** mBot2, dators (ar USB portu), USB-C vads, mBlock programmatūra

### Mērķis

- Iemācīties kā likt mBot2 sekot līnijām.
- Iemācīties kā likt mBot2 darīt konkrētas aktivitātes, balstoties uz krāsu norādījumiem.

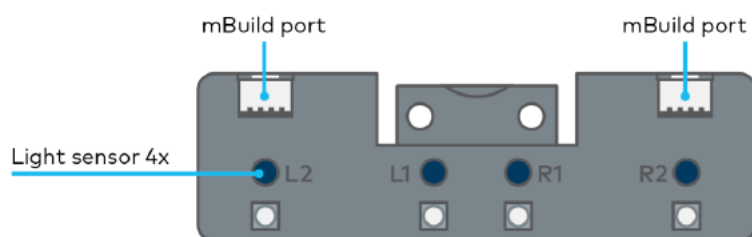
### Aktivitātes apraksts

*Iesildīšanās (5 minūtes)*

Krāsu sensori var noteikt gaismas intensitāti un krāsu. Jūs sastapsities ar šāda veida sensoriem daudzās dažādās ikdienas dzīves vietās. Piemēram, noliktavas robots var atrast pareizo maršrutu cauri noliktavai, pamatojoties uz dažādām grīdas krāsām. Šāda veida sensors tiek izmantots arī tad, ja nepieciešama precīza krāsu saskaņošana, piemēram, transportlīdzekļa krāsošanā. Mērierīce ar krāsu sensoru, krāsu mērītājs var pārbaudīt, vai transportlīdzekļa krāsošanā tiek izmantota pareizā krāsa. Palūdziet skolēniem iedomāties citus piemērus.

Krāsu sensors sastāv no trim dažādiem (iekšējiem) sensoriem, kas mēra objekta atstaroto gaismu, pamatojoties uz sarkanās, zaļās un zilās krāsas vērtību intensitāti. Tāpēc krāsu sensorus sauc arī par "RGB sensoriem". Jebkuru krāsu sensors atšifrē trīs vērtības, tāpēc konkrēts šo vērtību "mix" attēlo krāsu.

mBot2 ir četri no šiem RGB sensoriem, kas integrēti vienā sensorā. Apskatiet attēlu zemāk.



Šis ir Quad RGB sensors, kas atrodas mBot2. RGB sensori ir nosaukti L1, L2, R1 un R2 (L kreisajai pusei, R labajā pusē). Tie automātiski nosaka atspoguļotās krāsas RGB vērtības un iekšēji salīdzina vērtību sajaukumu ar dažādām krāsu iepriekš iestatītajām vērtībām. Tas ievērojami atvieglo programmēšanu, jo sensors pēc tam var ziņot par krāsu un jums nav jāpārbauda RGB programmu pašiem. Sensors var noteikt 6 dažādas krāsas, kā arī melnbaltu. Krāsas vietā tas var pārsegt pelēkā nokrāsā vai līnijās un krustojumos.

Quad-RGB sensors ļauj mBot2 sekot līnijai uz grīdas un reaģēt uz dažādiem krāsu marķieriem. Quad RGB sensors ir novietots mBot2 priekšpusē. Apakšējā virsmā redzēsiet četrus sensorus.

Augšpusē varat atrast katra nosaukumu, lai jūs varētu identificēt un programmēt sensorus noteiktā veidā. Apskatiet attēlu.



Kalibrēšanai tiek izmantota mazā poga sensora augšējā pusē. Veicot dubultklikšķi uz tā, gaismas diodes sāks mirgot, un jūs varat "pārvilkt" mBot2 ar sensoru pāri līnijai, kas jāievēro. Tas kalibrēs sensoru, lai atšķirtu līniju no fona. Ja izmantojat melnu līniju uz baltas virsmas, tas parasti nav nepieciešams.

Kartē, kas tiek nodrošināta ar mBot2, pašā celiņa iekšpusē ir atzīmētas krāsas, lai reaģētu uz krāsu kodiem. Tas var izraisīt sensora pārpratumu par spilgtu fona krāsu (piemēram, dzeltenu), nevis celiņu. Vispirms, lūdzu, kalibrējiet sensoru pēc dzeltenās krāsas koda, un tas automātiski atpazīs šo un visas "tumšākas" krāsas kā līniju. Programmējot mBot2, jūs izmantosiet vienu vai vairākus no šiem četriem sensoriem, lai pārbaudītu noteiktas krāsas uz grīdas. Varat likt mBot2 izpildīt komandu, kad tiek pamanīta krāsa. Piemēram, "apstāties", kad ir sarkans, un "iet pa kreisi", ja ir zaļš.

#### *Pirmo darbību apgūšana (10 minūtes)*

Programmā mBlock ir vairāki programmēšanas bloki, kurus varat izmantot četrkāršo RGB sensoru programmēšanai. Šos programmēšanas blokus varat atrast mBlock bloku lauka kategorijā "Quad RGB". Šie programmēšanas bloki ir zaļā krāsā.

Tālāk esošajā tabulā varat redzēt dažus programmēšanas blokus, kurus viņi var izmantot mūsu programmā ar Quad RGB sensoru. Pirmie četri programmēšanas bloki ir paredzēti (vienkāršai) līnijas identifikācijai. Tos var izmantot vienkāršai un progresīvai līniju sekošanai, kā arī krustojumu un asu (90 grādu) pagriezienu noteikšanai. Sensora kalibrēšana dažādu krāsu celiņiem ir īsi minēta iepriekšējā sadaļā. Zilās gaismas diodes sensora augšējā pusē norāda uz līnijas/fona noteikšanu: ja tās ir izslēgtas, tās uztver tumšu krāsu (līniju), ja tās spīd zilā krāsā, tās norāda uz spilgtu (fona) krāsu. Tāpēc gaismas diodes statuss norāda uz grīdas vai kartes atstarošanas spēju.

 quad rgb sensor 1 ▾ L1, R1's line ▾ status is (3) 11 ▾ ?

 quad rgb sensor 1 ▾ L1, R1's line ▾ status (0~3)

Šie programmēšanas bloki līnijas noteikšanā ņem vērā tikai divus iekšējos sensorus L1 un R1. Šie bloki ir paredzēti vienkāršai līniju sekošanai. Izmantojiet pirmo bloku nosacījumos. Izmantojiet otro bloku, ja vēlaties ziņot par ekrāna vērtību vai saglabāt to kā mainīgo.

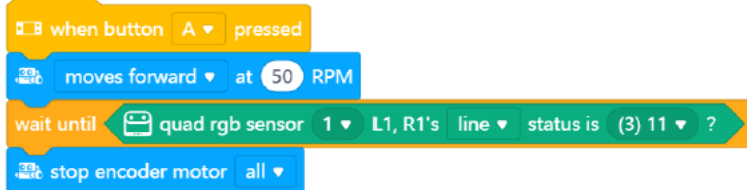
Ja sensors atrodas virs melnas līnijas, abām iekšējām gaismas diodēm jābūt izslēgtām. Tā kā katrs sensors (L1, R1) ar šo bloku var noteikt līniju vai fonu, ir 4 iespējamās kombinācijas: tās tiks attēlotas ar cipariem no 0 līdz 3 (uzziņai, melnais parāda arī bināros modeļus);

L1	R1	Nozīme	Status (decimāls)
		Sensors uz balta fona, abi LED ir ieslēgti	0
		R1 uztver melno līniju	1
		L1 uztver melno līniju	2
		Abi iekšējie sensori uztver melno līniju	3

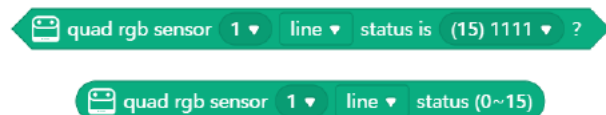
Atcerieties, ka gaismas diodes parāda fona stāvokli (ieslēgts vai loģisks "paties" = fons atklāts), savukārt šis programmēšanas bloks nosaka līnijas krāsu (loģiski "paties" - līnija noteikta).

Ja vēlaties izmantot vairāk nekā vienu Quad-RGB sensoru, varat norādīt programmai, kurus sensorus izmantot ar pirmo numuru visos turpmākajos programmēšanas blokos. Apskatiet attēlu zemāk.

Cipars 1 norāda pirmo pievienoto sensoru, numurs 2 otro pievienoto sensoru un tā tālāk. Nākamajā piemērā tiks izmantota šī bloka līnijas noteikšanas funkcija un apturēs robotu, tiklīdz tas nonāks pie melnas līnijas. Šis piemērs vēlāk tiks mainīts uz līnijas sekošanu.



Tā kā reakcijai braukšanas laikā var būt izšķiroša nozīme, ir svarīgi izmantot augšupielādes režīmu. Varat mēģināt pats un redzēt atšķirību, pārslēdzot režīmus.



Izmantojiet pirmo bloku nosacījuma izteikumos. Izmantojiet otro bloku, ja vēlaties ziņot par vērtību ekrānā vai saglabāt to mainīgajā.

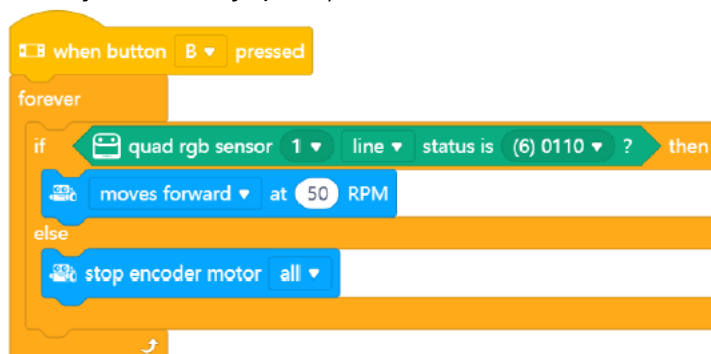
Šie bloki ievēro tieši tādu pašu loģiku kā iepriekšējais, izņemot to, ka tiem ir lielāks noteikšanas laukums un tie var identificēt krustojumus, vienlaikus sekojot līnijai. Atcerieties, ka bloki pārbauda līnijas statusu, tādēļ, ja viens no četriem sensoriem identificē (tumšu) līniju, atbilstošais binārais cipars tiks iestatīts uz "1" (loģiski paties), bet gaismas diodes nodzīsīs (neatspoguļo). šo sensoru no fona).

Tagad ar 4 atsevišķiem sensoriem diapazons palielinās līdz 16 dažādiem statusiem. Šajā tabulā ir parādīti visatbilstošākie statusi:

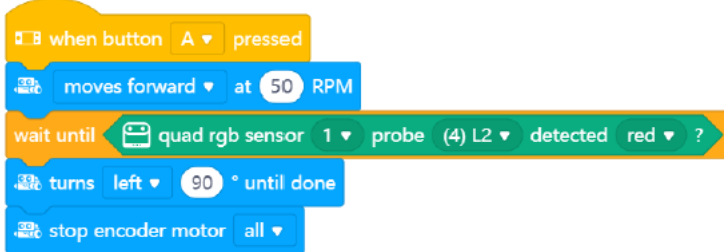
Binārais kods	Nozīme	Status (decimāls)
0000	Visi 4 sensori uz balta fona (nav ceļa)	0
0010	Tikai R1 uz melnās līnijas	2
0100	Tikai L1 uz melnās līnijas	4
0110	Iekšējie sensori R1 un L1 uz melnās līnijas	6
0111	Ass pagriezies pa labi (L1 un R1 uz melnās līnijas, krustojums pa labi, L-formas kurstoņums pa labi)	7
1110	Ass pagriezies pa kreisi (L1 un R1 uz melnās līnijas, krustojums pa kreisi, L-formas kurstoņums pa kreisi)	14
1111	T-formas krustojums (L1 un R1 uz melnās līnijas, krustojums pa labi un pa kreisi)	15

X formas krustojumu var noteikt, tikai braucot pāri T veida krustojumam un pārbaudot, vai viduslīnija turpinās.

Sekojošā piemēra programma liks robotam braukt tik ilgi, kamēr tas atrodas uz melnas līnijas ar diviem iekšējiem sensoriem. Tas apstāsies, ja tas noies no ceļa, tiks atklāts krustojums vai līnija pēkšņi beidzas:



Izmantojot šo programmēšanas bloku, jūs izvēlēsieties vienu no Quad RGB sensoriem (L2, L1, R1, R2) un noteicāt, vai tam jāatpazīst līnija, fons, balts, melns vai jebkura cita no 5 krāsām (sarkana, zaļa, zila, dzeltena), ciāna, violeta). Līniju noteikšanai un sekošanai ir ieteicami iepriekšējie bloki, jo tie vienlaikus pārbauda vairāk nekā vienu sensoru (ātrāk, precīzāk, jo robots nepārvietojas starp rādījumiem). Bet jūs varat arī izmantot šo bloku, lai noteiktu, piemēram, sarkanās krāsas tirgu robota kreisajā pusē, kamēr robots seko līnijai. Programmēšanas piemērā, ja nospiežat pogu B, mBot2 brauc taisni (pieņemot, ka tas seko līnijai) un veic 90 grādu pagriezienu, kad (iedomātā) celiņa kreisajā pusē tiek konstatēts sarkans marķieris.



Šis programmēšanas bloks mēra, cik tālu robots atrodas no melnās trases. Varat to izmantot progresīvai un vienmērīgai līniju sekošanai. Iepriekšējā rindā, kas seko programmēšanas blokam, tiek izmantota trīskārša atšķirība: uz ceļa, pa kreisi vai pa labi (atsevišķos gadījumos pievienota krustojuma noteikšana). Bet ikdienā, braucot ar velosipēdu, stūri precīzāk pielāgotu līkumam. Šis bloks pieļauj proporcionālu līniju: jo vairāk robots atrodas ārpus trases, jo vairāk tas stūrēs pretējā virzienā. Ja nobīde no trases ir 0, stūrēšana ir 0, kā arī (taisni).

Tālāk redzamajā programmas paraugā ekrānā tiks parādīta novirze, tāpēc vispirms varat mēģināt pārvietot robotu uz izsekošanas ar roku:



Iepriekš redzamajā diagrammā katram Quad RGB sensora koda blokam ir programmēšanas piemērs. Studenti gatavojas atkārtoti izveidot šos programmēšanas piemērus mBlock un pārbaudīt tos. Vienam programmēšanas piemēram palūdziet viņiem pašiem izstrādāt koda modifikācijas.

mBot2 gatavojas sekot trasei kartē, būdams tūrisma treneris. Šim nolūkam jums būs jāizmanto līdz šim izskaidrotie un pārbaudītie kodu piemēri un tie attiecīgi jāmaina.

No sākuma lietojiet šo programmēšanas bloku:



Ja robots atrodas trasē (gan L1, gan R1 nosaka līniju, koda vērtība 3), robotam jābrauc taisni. Ja tikai viens no diviem iekšējiem sensoriem identificē līniju, tam attiecīgi jāpagriežas pa kreisi vai pa labi.

Izmantojiet šos braukšanas blokus (tiek rādīts tikai pagrieziens pa labi):



Viņi vienkārši iedarbina motorus pēc vajadzības, un nākamais koda bloks tiks izpildīts nekavējoties.

Tā kā viņi nezina, cik tālu robots ir izslēgts vai kad notiks nākamais līkums, braukšanas komandas nevar ietvert braukšanas laiku vai attālumu (tas pats attiecas uz pagriezienu). Nākamajā programmas cilpas atkārtojumā pārbaudīs jaunus sensora datus un attiecīgi izvēlēsies braukšanu.

Ja izmantojat pievienoto karti, noteikti kalibrējiet sensoru, kā aprakstīts, spilgtākajā krāsā (dzeltenā sadaļa trasē).

### *Izmēģināšana (25 minūtes)*

Viņi tagad izveidos savu programmu mBlock, izmantojot RGB sensoru.

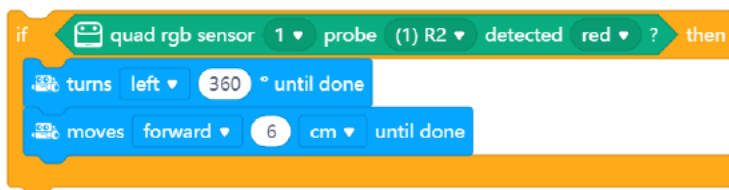
Šajā uzdevumā viņi mBot2 pārvērtīs par tūristu automašīnu. Ekskursijas treneris dodas ekskursijā pa pilsētas orientieriem. mBot lodziņā ir karte ar trasi. Treneris seko melnajai līnijai kartē. Iepriekšējā daļā viņiem jau vajadzēja ieprogrammēt vienkāršu līnijas sekošanas robotu, nu ir pienācis laiks pievienot papildu uzlabojumus.

Lai programma būtu vieglāk saprotama, jūs varat "uzbūvēt" savus programmēšanas blokus. Visu programmas daļu, kas seko rindai, var pievienot līdz vienai vienkāršai komandai, piemēram, "vienkāršā rindīņa". Sarkanā/rozā "Mani bloki" koda izvēlnē ļauj definēt pielāgotu bloku un jaukā veidā strukturēt programmu. Tālāk esošajā programmā piemērā tiek izmantoti pielāgoti bloki.

Papildus trases sekošanai treneris veic darbību dažos apskates objektos. Apskates objekti ir norādīti ar krāsainu laukumu (sarkanu, dzeltenu, zaļu un zilu).

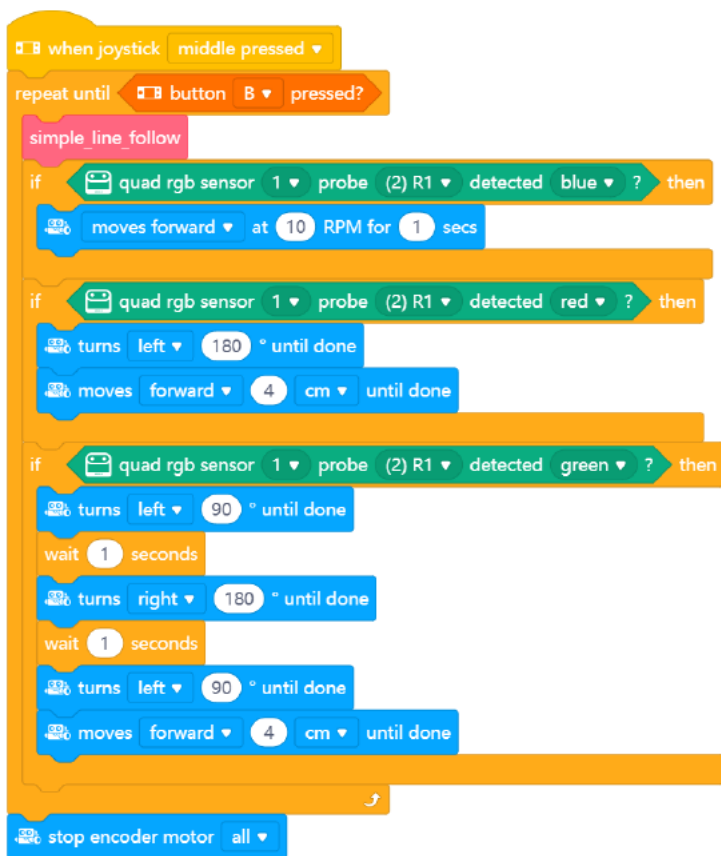
Jūs varat izdomāt darbības, kas trenerim būtu jāveic. Vai jūs gatavojaties autobusam braukt pa Parīzi? Tad varētu būt jautri spēlēt Francijas himnu. Vai arī Bigbena zvani, ja treneris brauc pa Londonu.

Protams, jūs varat arī padarīt to nedaudz vienkāršāku. Tālāk esošajā piemērā trenerim jāgriežas pie sarkanās atzīmes un jāveic dažādas darbības pie citiem krāsainiem marķieriem.



```
if quad rgb sensor 1 probe (1) R2 detected red ? then
  turns left 360 until done
  moves forward 6 cm until done
```

Izmantojiet iegūtās zināšanas, protams, jūs varat daudz eksperimentēt ar dažādām programmēšanas iespējām mBlock. Vai jūs iestrēgāt? Pēc tam varat izmantot tālāk norādītos programmēšanas piemērus. Varat to kopēt un pielāgot savām vēlmēm un vēlmēm.



```
when joystick middle pressed
  repeat until button B pressed?
  simple_line_follow
  if quad rgb sensor 1 probe (2) R1 detected blue ? then
    moves forward at 10 RPM for 1 secs
  if quad rgb sensor 1 probe (2) R1 detected red ? then
    turns left 180 until done
    moves forward 4 cm until done
  if quad rgb sensor 1 probe (2) R1 detected green ? then
    turns left 90 until done
    wait 1 seconds
    turns right 180 until done
    wait 1 seconds
    turns left 90 until done
    moves forward 4 cm until done
  stop encoder motor all
```

Šī programma sākas, ja nospiežat B, un tā apstājas, nospiežot A. Tā izpilda vienkāršu rindiņu, izmantojot programmu "Mani bloki", un pēc tam pārbauda, vai R1 sensors nosaka vai nu zilu (braukt ļoti lēni), sarkanu (apgriezties) vai zaļu. (skatieties uz katru pusi). Lai pārlicinātos, ka krāsa pēc darbību veikšanas vairs netiek atpazīta, robots katru reizi brauc 4 cm uz priekšu.

Bloks "simple\_line" ir definēts šādi:



```
define simple_line_follow
if quad rgb sensor 1 L1, R1's line status is (3) 11 ? then
  moves forward at 50 RPM
if quad rgb sensor 1 L1, R1's line status is (1) 01 ? then
  turns right at 20 RPM
if quad rgb sensor 1 L1, R1's line status is (2) 10 ? then
  turns left at 20 RPM
if quad rgb sensor 1 L1, R1's line status is (0) 00 ? then
  moves backward at 10 RPM
```

*Noslēgums (5 minūtes)*

Vai sanāca paveikt uzdevumu un izveidot tūrista maršrutu?

Nodarbības beigās pārrunājiet šādus jautājumus:

- Kas sanāca labi?
- Kas varētu būt labāk?
- Kura nodarbības daļa bija visvieglākā un kura bija visgrūtākā?
- Par kurām lietām vēl būtu nepieciešams vairāk izrunāt/apskatīt?

## 6. Pavasara dabas atklāšana

**Autors:** Tartu bērnudārzs Kannike

**Sarežģītums:** Iesācēji

**Vecums:** 3-4

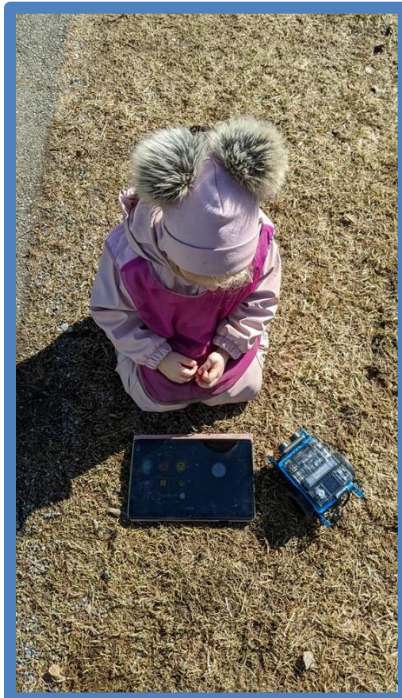
**Nepieciešamie resursi:** mBot2, planšetdators, palielināmais stikls vai citas ierīces pētīšanai, [Makeblock aplikācija](#)

### Mērķis

- Dabā atrodošo augu un kukaiņu atklāšana.
- Programmēšanas pamati.

### Aktivitātes apraksts

Mēs izmantojam mBot2, lai atklātu pavasara dabu. Bērni ar robotu iepazīstināja dažādus dabas objektus (zarus, čiekurus, zīles, kokus, kukaiņus u.c.). Kopā ar robota palīdzību mēģinājām savākt dažādus dabas objektus un no tiem izveidot kukaiņu ligzdas. Bērni apsprieda, kuri kukaiņi varētu dzīvot mūsu ligzdā, un, kamēr viens bērns ar robotu veidoja ligzdu, pārējie meklēja kukaiņus un spiegoja dabu no dažādiem leņķiem, un tas viss pēc kārtas.



# 7. Aplikācijas izzināšana

**Autors:** Tartu bērnudārzs Kannike

**Sarežģītums:** Iesācēji

**Vecums:** 5-6

**Nepieciešamie resursi:** mBot2, planšetdators, palielināmais stikls vai citas ierīces pētīšanai, [Makeblock aplikācija](#)

## Mērķis

- Dabā atrodošo augu un kukaiņu atklāšana.
- Programmēšanas pamati.

## Aktivitātes apraksts

Izmantojot [Makeblock aplikāciju](#) izzinājām mBot2 un iepazīstinājām ar dažādām iespējām. Eksperimentēja ar dažādām kustībām, vizuālajām un skaņas iespējām, piemēram spēlējām klavieres.

