

# mBot robot ringrajal

Veiko Vunder

September, 2017

## 1 Ülesande kirjeldus

Robot nimega "mBot" (pildil 1) soovib sõita vastupäeva mööda ringjoont, mille läbimõõt  $d = 100$  cm. On teada, et rataste keskkohad asuvad üksteisest  $a = 64$  mm kaugusel ja robot soovib sõita nii, et ringjoon paikneks koguaeg täpselt kahe ratta vahel. Et seda saavutada, peab mBot koguaeg natuke pöörama, mis tähendab, et roboti rattad on vaja panna pöörlema erineva kiirusega. Kuid missugused kiirused valida?

### Ülesanne 1

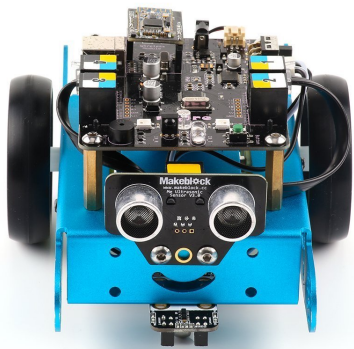
Sinu esimene ülesanne on leida jagatis  $k$ , mis näitab, kui palju peab vasak ratas pöörlema paremast aeglasemalt.

### Ülesanne 2

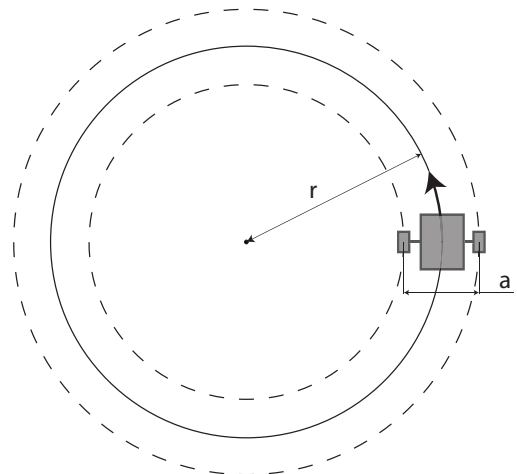
Tähistagu  $v_p$  parema ratta kiirust. Leia avaldis vasaku ratta kiiruse  $v_v$  arvutamiseks, kasutades kasutades eelnevalt leitud jagatist  $k$ .

### Ülesanne 3

Tutvu peatükiga 2 ja programmeeri mBot ringjoont mööda sõitma nii, et arvutused teeks robot.



Joonis 1: mBot roboti eestvaade.



Joonis 2: mBot ringjoonel.

## 2 Roboti programmeerimine

Järgnevalt anname kiire ülevaate, kuidas kasutada Arduino IDE keskkonda [1] mBot roboti programmeerimiseks.

### 2.1 Arduino IDE

Kui oled avanud uue Arduino lähtefaili (*File* ⇒ *New*), siis leiad sealt kaks funktsiooni: *setup()* ja *loop()*. Kui robot sisse lülitada, siis hakatakse programmi järjekorras ülevalt alla täitma. Kõigepealt käivitatakse üks kord *setup()* funktsiooni sees asuvad käsud ja seejärel **korraldatakse lõpumatult** *loop()* funktsioonis olevaid käsked. Seega, kui soovid, et robot teeks midagi ainult üks kord, siis kirjuta käsud *setup()* funktsiooni sisse. Aga kui soovid roboti programmeerida midagi tegema pidevalt (näiteks võrdlema kaugusanduri näitu), siis sobib selleks *loop()*. Lähtekoodi saab robotile sobivaks masinkoodiks teha nupuga (**Verify**) ja saata robotisse nupuga (**Upload**). Vaata videost [3], kuidas teha õiged valikud menüüdes *port* ja *board*.

### 2.2 Arduino IDE ja mBot

Arduino IDE'ga saab programmeerida väga erinevaid seadmeid. Igal seadmel on oma kindel viis, kuidas sellega infot vahetada või seda juhtida. Algajast robotihuviline ei taha kindlasti kõike seda keerulist bittide ja baitide kodeerimist ise teha, vaid soovib anda seadmetele lihtsaid käsked:

LCD ekraanile: "joonista vasakusse ülesse nurka täht A"

kaugusandurile: "kui kaugel asub esimene lähim takistus?"

mootorile: "pöörle 20% kiirusega"

mBot-i jaoks on hulganisti sarnaseid lihtsaid käsked juba tootja poolt valmis programmeeritud [2]. Nende kasutamiseks, on vaja programmi algusesse lisada järgmine rida:

```
#include "MeMCore.h"
```

Nüüd saame teha kummagi mootori jaoks objektid, mille lisame enne *setup()* funktsiooni algust:

```
MeDCMotor vasakMootor(M1);  
MeDCMotor paremMootor(M2);
```

M1 ja M2 näitavad, mis pessa ühendatud mootorit me selle objektiga juhtida soovime. Mootori võimsust saab määrata meetodiga **run(KIIRUS)**, kus sulgude sisse kirjutame täisarvu vahemikus -255...255. Näiteks:

```
vasakMootor.run(120);  
paremMootor.run(-120);
```

Nii pannakse mootorid käima sama kiirusega, aga eri suunas. Pöörlemise suuna määrab see, kas arv on positiivne või negatiivne. Kokkuvõtlikult:

- -255: mootor pöörleb maksimaalse kiirusega vastupäeva;
- 0: mootor seisab;
- 255: mootor pöörleb maksimaalse kiirusega päripäeva.

### 2.3 mBot ja aeg

Eelnevaga saame roboti liikuma, aga kuidas teha nii, et robot ise 2 sekundi pärast seisma jääks? Siin on abiks funktsioon **delay(AEG\_MILLISEKUNDITES)**. Programmi täitmine ei jätku enne, kui sulgude sees täpsustatud aeg on täis tiksunud. Näide *delay()* kasutamisest:

```
vasakMootor.run(50);  
paremMootor.run(-50);  
delay(2000);  
vasakMootor.run(0);  
paremMootor.run(0);
```

Kui `delay()` vahelt ära jätta, siis mootorid lülitatakse enne välja kui nad jõuavad üldse pöörlema hakatagi. Kas teadsid, et kiirematel meist kulub silma pilgutamiseks 0,1 sekundit ehk 100 millisekundit? Ülal toodud käske täidetakse aga veel umbes 1000 korda kiiremini ehk mikrosekundite skaalas. Seega võid ettekujutada, et ühe sinu silmapilgu jooksul on robotil aega neid nelja rida korrata üle 1000 korra. Seetõttu ei märka me ka erinevust, et vasak mootor pandi tegelikult käima pisut enne kui parem.

## Viited

- [1] *Arduino - Software*. URL: <https://www.arduino.cc/en/Main/Software> (Viimati külastatud: 01 Sept. 2017).
- [2] Makeblock. *Makeblock-Libraries: Arduino Library for Makeblock Electronic Modules, learn more from Makeblock official website*. Sept. 1, 2017. URL: <https://github.com/Makeblock-official/Makeblock-Libraries>.
- [3] *Tartu Ülikooli online kursuse "Robootikast puust ja punaseks" õppevideo Video 3.02. Arduino programmeerimiskeskonna olulisemad osad*. URL: <https://sisu.ut.ee/robot/31-arduino-arenduskeskkond> (Viimati külastatud: 01 Sept. 2017).

### 3 Lahenduskäik

Paigutame roboti ringjoonele raadiusega  $r$  ja tähistame roboti rataste vahekauguse  $a$  (Joonis 1). Vasaku ja parema ratta teepikkusteks saame vastavalt

$$s_v = 2\pi(r - a/2) \quad (1)$$

$$s_p = 2\pi(r + a/2) \quad (2)$$

Olgu  $k$  jagatis, mis näitab, mitu korda on vasaku ratta teepikkus parema teepikkusest lühem.

$$k = \frac{s_v}{s_p} = \frac{(r - a/2)}{(r + a/2)} \quad (3)$$

Kuna mõlemad rattad läbivad teepikkused sama ajaga, siis peavad nende kiirused olema  $k$  korda erinevad. Mis tähendab, et kui valime roboti parema ratta kiiruseks  $v_p$ , siis vasaku ratta kiirus on  $v_v = v_p k$ .