Robot *mBot* väldib takistusi

Veiko Vunder

September, 2017

1 Sissejuhatus

Selle tunni eesmärgiks on programmeerida robot nimega *mBot* (pildil 1) kasutama ultraheli kaugusandurit ja selle abil vältima asjadele otsasõitmist.

2 Arduino IDE

Siin peatükis saad kiire ülevaate Arduino IDE arenduskeskkonnast[1] ja õpid, kuidas teha vajalikud ettevalmistused mBot roboti programmeerimiseks.

Ava Arduino IDE keskkond ning seejärel loo uus Arduino lähtefail (***File*** ⇒ ***New***). Avanenud aknast leiad kaks tühja plokki: ***setup()*** ja ***loop()***. Programmeerimismaailmas nimetatakse selliseid plokke funktsioonideks. Funktsioonide abil saab muidu suure ja keerulise programmi jagada väiksemateks alamosadeks ja neid osi korduvalt kasutada.

Käsud, mis on kirjutatud *setup()* funktsiooni, käivitatakse **üks kord** roboti sisselülitamisel. *loop()* funktsioonis olevaid käske täidetakse uuesti ja uuesti kuni robot välja lülitatakse. 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()*.

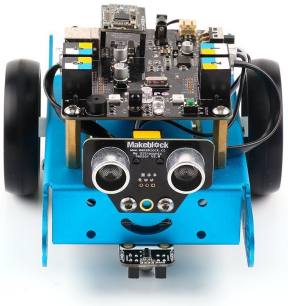
Nüüd oleme valmis robotit arvutiga ühendama.

Tähelepanu! Kui sa pole kindel, mis programm on viimati robotisse laetud, siis selleks, et mBot ootamatult koos juhtmetega minema ei kihutaks, tasub robotit esialgu hoida käes või asetada lauale rattad ülespidi!

Ühenda robot arvutiga ja lülita robot sisse. Järgi videos [3] jagatud õpetusi, et teha õiged valikud menüüdes ***Tools*** ⇒ ***Board*** ja ***Tools*** ⇒ ***Port***.

Testime, kas robot on ühendatud ja eelnevad valikud õiged. Selleks teeme praeguse tühja programmi robotile sobivaks masinkoodiks ehk kompilleerime programmi ***Sketch*** ⇒ ***Verify/Compile***. Oota kuni olekuribale ilmub *"Done Compiling"* (varu kannatust, esimesel korral tuleb pisut kauem oodata). Järgnevalt saadame programmi robotisse ***Sketch*** ⇒ ***Upload*** ja ootame, et olekuribale ilmuks *"Done Uploading"*. Programmi kompilleerimist ja robotisse laadimist saab teha ka nuppude abil (pildil 2).

Kui kõik läks hästi, siis Arduino IDE on programmeerimiseks valmis seatud ja robotisse on paigaldatud programm, mis ei tee mitte midagi. Parandame selle vea järgmises peatükis 3.



Joonis 1: mBot roboti eestvaade.

Joonis 2: Nupud programmi kompilleerimiseks ja robotisse laadimiseks.

3 mBot programmeerimine

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 käske nagu "joonista LCD ekraani vasakusse ülesse nurka täht A", "kui kaugel asub takistus?" või "pane mootor päripäeva pöörlema 20% kiirusega".

Kõik eelpool mainitud käsud ja veel palju muudki on mBot-i jaoks juba tootja poolt valmis programmeeritud [2]. Nende käskude kasutamiseks, on vaja lisada programmi algusesse (enne *setup()* funktsiooni) järgmine rida:

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

3.1 Mootorid pöörlema

Selleks, et juhtida mBoti mootoreid, on vaja kummagi mootori jaoks teha objektid. Lisame pärast *#include "MeMCore.h"*, aga enne *setup()* funktsiooni järgnevad read:

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

Märksõna *MeDCMotor* ütleb, et soovime teha mBot mootori objekti. *vasakMootor* ja *paremMootor* on mootoritele antud nimed ja sulgudes on vaja täpsustada, millisesse pessa on mootor ühendatud.

Mootori kiiruse seadmiseks on mBot-i arendajad programmeerinud funktsiooni **run(*KIIRUS*)**, kus sulgude sisse täpsustame kiiruse täisarvuna vahemikus $-255 \ldots 255$. Järgnevad kaks rida panevad mootorid pöörlema keskmise kiirusega:

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

Kirjuta eelnevad kaks rida *setup()* funktsiooni sisse (loogeliste sulgude vahele), kompilleeri programm ja saada robotisse. Jälgi, mis suunas hakkavad roboti rattad pöörlema. Mis suunas robot liigub, kui see hetkeks ratastele asetada?

Mootori pöörlemise suuna määrab see, kas arv on positiivne või negatiivne. Kokkuvõtlikult:

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

Katseta erinevate kiirustega ja programmeeri robot liikuma otse nii aeglaselt kui võimalik.

3.2 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.

3.3 Sõitmise funktsioonid ja kaugusandur

Ava näitefail ***File*** ⇒ ***Sketchbook*** ⇒ ***mbot_soidab_takistuseni***, kompilleeri ja lae robotisse.

Selles programmis on tehtud objekt kaugusanduri kasutamiseks ja lisatud kolm funktsiooni sõitmiseks. Otsi üles rida, kus tehakse kaugusanduri objekt. Kontrolli, et andur oleks samas pesas, mis programmis kirjutatud.

Kuidas määratakse selles programmis mootorite kiirus ja missuguse käsuga saadakse kaugusanduri näit?

Uuri, mis käsud käivitatakse ***loop()*** funktsiooni sees ja kuidas kasutatakse programmis sõitmise funktsioone ***otse()*** ja ***peatu()***.

Ülesanded

1. Muuda programmi nii, et robot jääks seisma, kui takistus asub **lähemal kui 20 cm**.
2. Muuda programmi nii, et robot **tagurdaks**, kui takistus asub lähemal kui 20 cm.

Lisaülesanded*

1. Uuri, mis suunas peavad rattad liikuma, et robot pööraks. Kirjuta sarnaselt *otse()*, *peatu()*, *tagurda()* funktsioonidele juurde funktsioonid ***vasakule()*** ja ***paremale()***.
2. Kasuta eelmises ülesandes loodud funktsioone nii, et kui robot on tuvastanud eseme, mis asub **lähemal kui 20 cm**, siis ta mitte ei seisa paigal, vaid **pöörab 0,5 sekundit**, et takistusest mööda hiilida.
3. Muuda programmi nii, et robot:
   - sõidaks edasi, kui takistus on kaugemal kui 20 cm;
   - peatuks kui takistus on lähemal kui 20 cm, aga kaugemal kui 10 cm;
   - tagurdaks, kui takistus on lähemal kui 10 cm.

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 programmeerimiskeskkonna olulisemad osad*. URL: `https://sisu.ut.ee/robot/31-arduino-arenduskeskkond` (Viimati külastatud: 01 Sept. 2017).