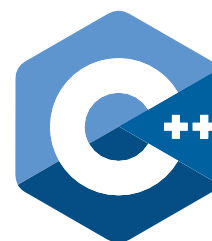
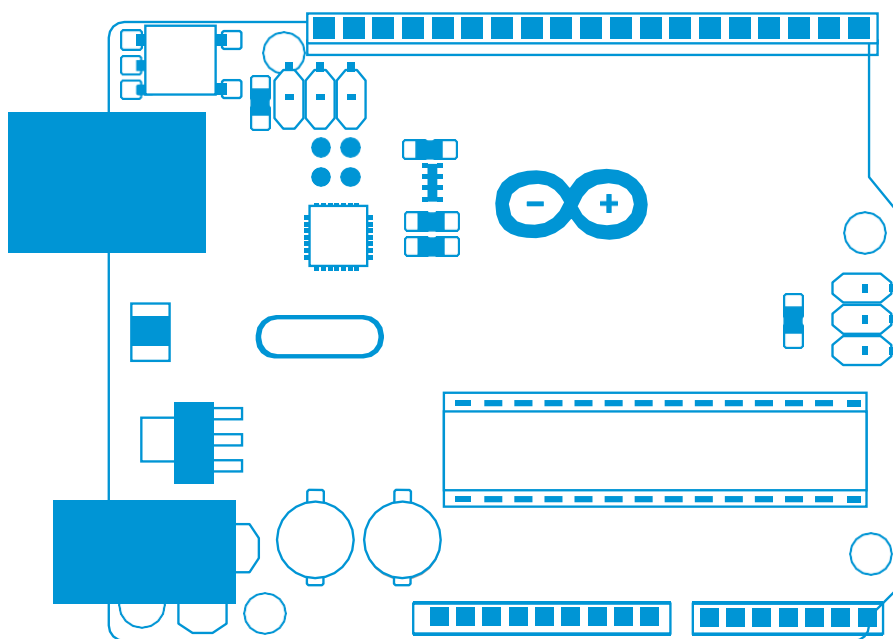
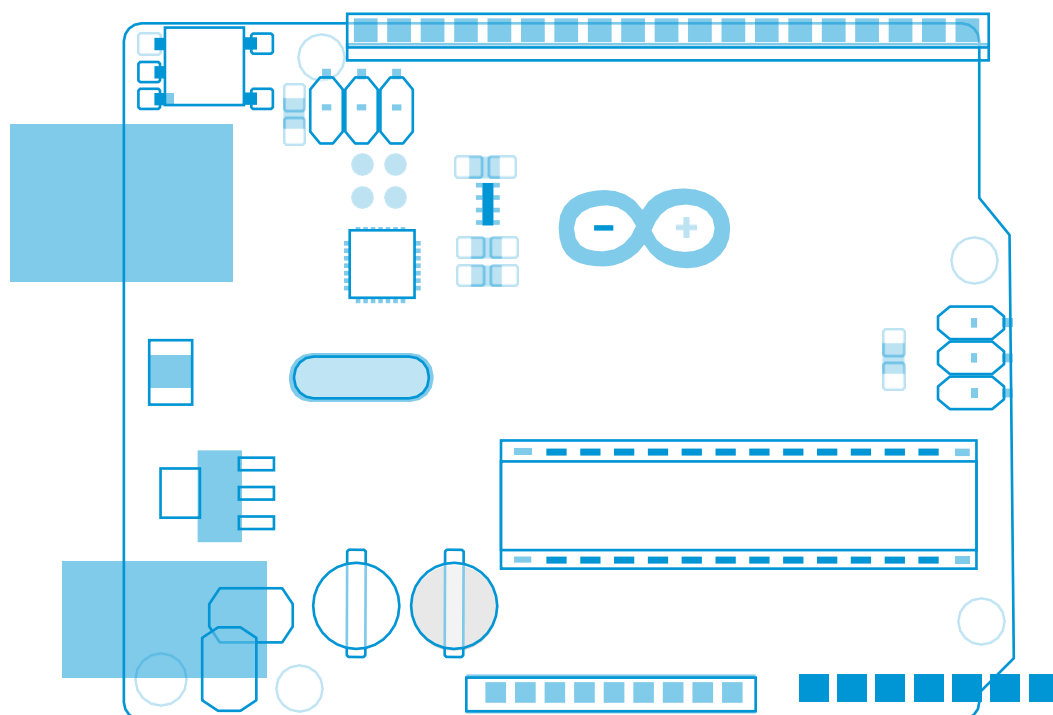
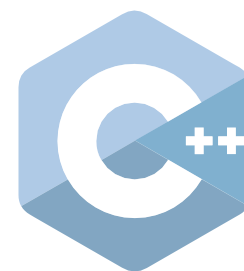


Kosmos on õpetlik

→ TUTVUME ARDUINOGA!

Sissejuhatus Arduino programmeerimisse C++ keeles





Tutvume arduinoga	lk 3
Tegevused – kirjeldus ja õpiväljundid	lk 4
Tegevus 0: Alustamine	lk 5
Tegevus 1: Pane mind vilkuma!	lk 8
Tegevus 2: SOS-i vilgutamine	lk 11
Tegevus 3: Temperatuuri mõõtmine	lk 12
Tegevus 4: Rõhu mõõtmine	lk 16
Tegevus 5: Kõrguse mõõtmine	lk 18
Töölehed õpilastele	lk 19
Lingid	lk 34

kosmos on õpetlik – tutvume arduinoga
www.esa.int/education

ESA Haridusosakond ootab tagasisidet ja kommentaare
education@esa.int (inglise keeles) või esero@ut.ee (eesti keeles)

Materjali koostas ESA Education

Autoriõigus 2018 © Euroopa Kosmoseagentuur

→ TUTVUME ARDUINOGA!

Sissejuhatus Arduino programmeerimisse C++ keeles

Lühiülevaade

Vanuserühm: 14 kuni 20 aastat

Seotud õppekavad: programmeerimine, elektroonika

Raskusaste: keskmine

Õppetunni kestus: 90 – 20 minutit

Asukoht: sisetingimustes

Vajalikud vahendid: Arduino tarkvara ja riistvara

Märksõnad: Arduino, andur, kood

Kirjeldus

Õpilased õpivad kasutama tehnoloogiaid, mis on kasutusel kosmoses Arduino seadme abil. Nad ehitavad voluringe, mille abil saab panna vilkuma valgusdioode ning mõõta temperatuuri, rõhku ja kõrgust. Õpilased tutvuvad programmeerimise algtõdedega C++ keeles Arduino IDE (integreeritud arenduskeskkond) tarkvara abil. Käesolevad tegevused on heaks lähtepunktiks, et tulevikus osaleda [CanSat võistlusel](#).

Õpieesmärgid

- Parandada analüüsivõimet
- Kujundada arusaam programmeerimiskeelest
- Kujundada arusaam voluringide koostamisest
- Mõista, kuidas andureid saab andmete kogumiseks kasutada
- Parandada meeskonnatöö oskuseid

Tegevused – kirjeldus ja õpiväljund

	Pealkiri	Kirjeldus	Õpiväljund	Nõuded	Aeg
0	Alustamine	Komponentide tutvustus, mida tegevustes kasutatakse	Õpilased tutvuvad komponentide ja nende funktsioonidega, mida nad kasutama hakkavad	Põhiteadmised elektroonika-komponentidest	10 minutit
1	Pane mind vilkuma!	Õpilased ehitavad oma esimese Arduino voluringi	Õpilased suudavad koodi abil valgusdiodi vilgutada, mille nad on kirjutanud	Eelmise tegevuse lõpuleviimine	15 minutit
2	SOS-i vilgutamine	Õpilased õpivad keerukamate teadete saatmist valgusdiodide abil	Õpilased kirjutavad koodi ja programmeerivad valgusdiodi SOS teadet edastama	Eelmiste tegevuste lõpuleviimine	15 minutit
3	Temperatuuri mõõtmine	Õpilased kasutavad termistori, et mõõta ümbritseva keskkonna temperatuuri	Õpilased suudavad ehitada keerukamaid voluringe, mille abil ümbritsevas keskkonnas mõõtmisi läbi viia	Eelmiste tegevuste lõpuleviimine	20 minutit
4	Rõhu mõõtmine	Õpilastele näidatakse, kuidas kasutada rõhuandurit õhurõhu mõõtmiseks	Õpilased suudavad ehitada keerukamaid voluringe, mille abil ümbritsevas keskkonnas mõõtmisi läbi viia	Eelmiste tegevuste lõpuleviimine	20 minutit
5	Kõrguse mõõtmine	Õpilastele näidatakse, kuidas õhurõhu mõõtmiste tulemuste abil kõrgus määrata	Õpilased suudavad juba enesekindlalt voluringe ehitada ja kirjutada koodi, et viia läbi ümbritsevas keskkonnas mõõtmisi	Eelmiste tegevuste lõpuleviimine	20 minutit

→ Tegevus 0: Alustamine

Sissejuhatus

Arduino on avatud lähtekoodiga platvorm, mida saab kasutada erinevate elektroonikaprojektide ehitamiseks, sealhulgas ka päris kosmosemissioonide simuleerimiseks nagu näiteks ExoMars (<http://exploration.esa.int/mars/>). Arduino koosneb füüsilisest programmeeritavast arendusplaadist (riistvara) ja tarkvarast ehk IDE-st (integreeritud arenduskeskkond), mis töötab arvutis.

Arduino riistvara ja tarkvara loodi kunstnike, disainerite, asjaarmastajate, häkkerite ja inimeste jaoks, keda huvitab interaktiivsete esemete ja keskkondade loomine. Arduino külge saab ühendada nuppe, valgusdioode, andureid, mootoreid, kõlareid, GPS seadmeid, kaameraid, selle saab ühendada internetiga ja ka teie nutitelefoni või televiisoriga! Tänu seadme paindlikkusele ning sellele, et Arduino tarkvara on tasuta, trükkplaadid ja andurid on suhteliselt odavad ja nii tarkvara kui ka riistvara on intuiitsed on sellel väga palju kasutajaid (www.arduino.cc).

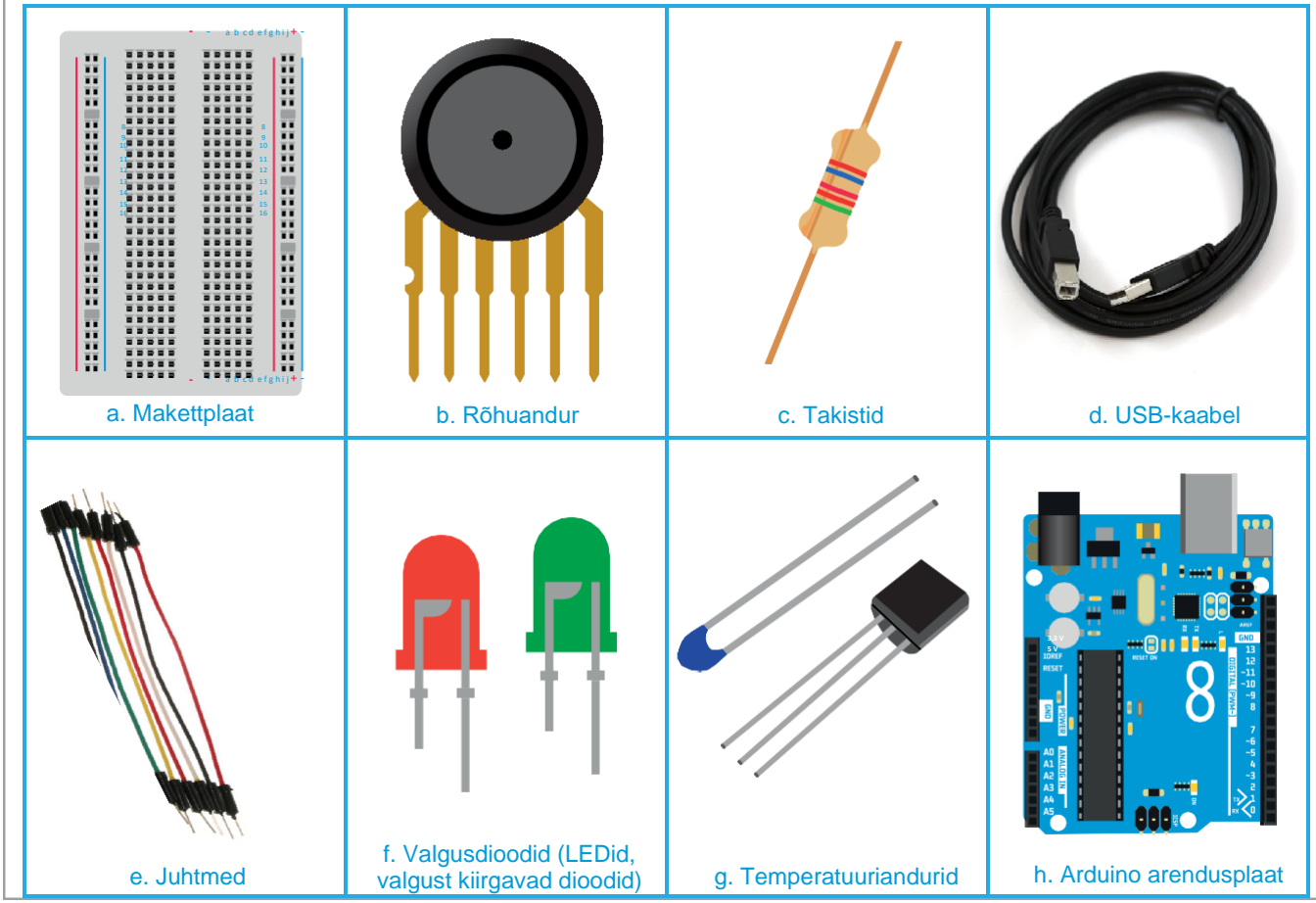
Riistvara

Selle tegevuse jaoks kasutame Arduino Unot (Joonis A1-h). Arduino suudab oma pesades (metallist jalad) pinget muuta ja lugeda nende pinget. Pane tähele, et kõik pesad, mille kõrval on tähis **GND** (maandus) on negatiivsed poolused. **Vin** või **5V** on positiivsed poolused (nagu aku positiivne ja negatiivne terminal).

Kasutame komponente, mis on esitatud Joonisel A1:

- a. **Makettplaat.** Tugiplaat selleks, et elektroonikakomponente oleks lihtne ühendada.
- b. **MPX4115A rõhuandur.** Komponent, mis mõõdab keskkonna absoluutrõhku.
- c. **Takistid väärtusega 220 Ω ja 10 kΩ.** Elektrivoolu juhtimiseks. Takistust mõõdetakse oomides (Ω). Takistitel on värvilised triibud, mis kujutavad takisti väärtust.
- d. **USB-kaabel.** Vajalik selleks, et ühendada Arduino arvuti ja/või toiteallikaga.
- e. **Juhtmed.** Vajalikud selleks, et komponente elektriliselt ühendada.
- f. **Valgusdioidid (LEDid, valgust kiirgavad dioidid).** Elektroonikakomponent, kus tekib voolu läbimisel valgus.
- g. **Temperatuuriandurid.** Analoogandur (nagu näiteks LM35) ja termistor (takisti, mis on temperatuuri suhtes eriti tundlik).
- h. **Arduino arendusplaat.** Töötab nagu lihtne arvuti, mis koosneb protsessorist, mälust ja sisenditest ning väljunditest.

Joonis A1



a. Makettplaat

b. Rõhuandur

c. Takistid

d. USB-kaabel

e. Juhtmed

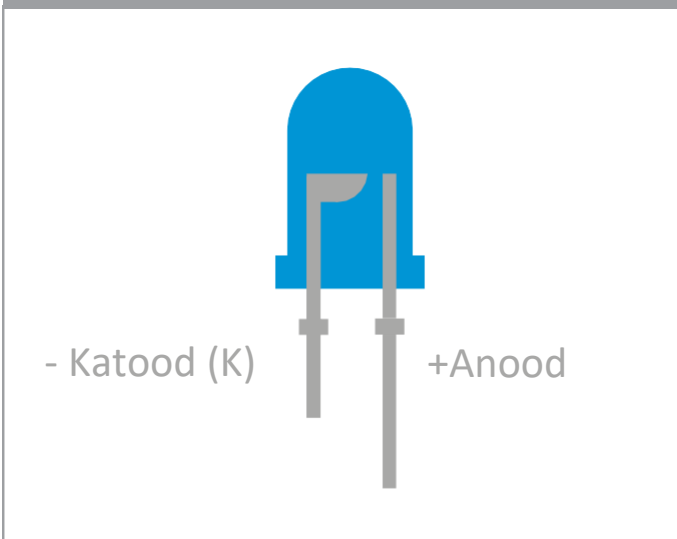
f. Valgusdiodid (LEDid, valgust kiirgavad diodid)

g. Temperatuuriandurid

h. Arduino arendusplaat

↑ Arduino põhikomponendid

Joonis A2

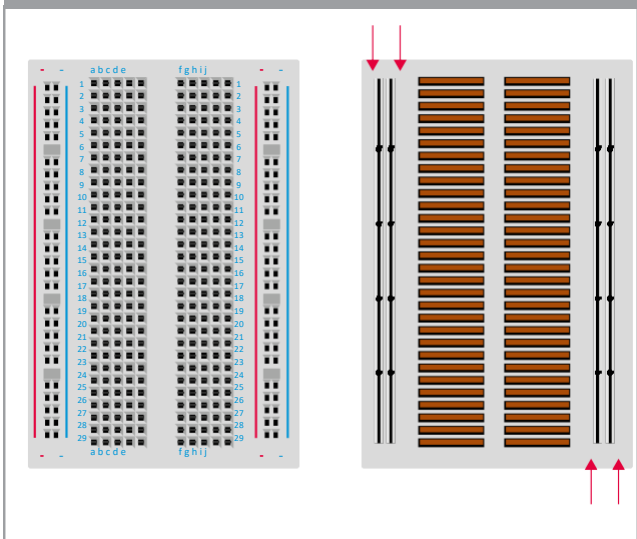


- Katood (K)

+Anood

↑ Valgusdiod. Pane tähele, et valgusdiodil on üks pikk ja üks lühike jalg. Pikka jalgat kutsutakse anoodiks ja lühikest katoodiks.

Joonis A3



↑ Makettplaat. Iga rea kõik pesad on omavahel ühendatud.

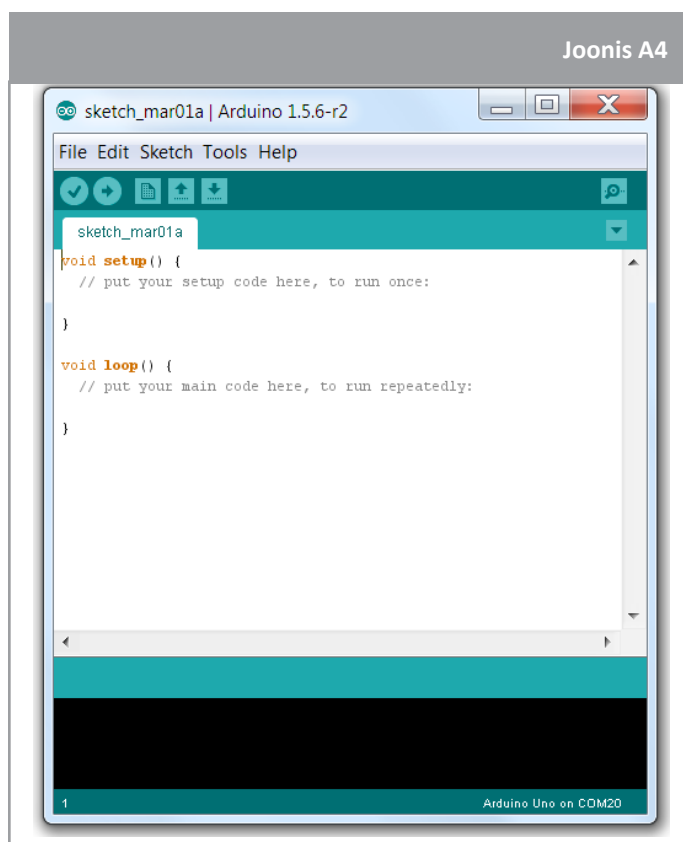
Kui Arduino ühendada arvutiga, peaks arvuti selle ära tundma ja paigaldama vajalikud draiverid. Kui draiverite paigaldamisega esineb probleeme, siis täpsemad juhised leiab siit: www.arduino.cc/en/Guide/ArduinoUno

*Palun vali õige arendusplaat (proovi valida Genuino Uno) ja õige jadaliides (COM) tööriistade menüüst!

Tarkvara







Arduino tarkvara saab kergesti alla laadida ja paigaldada siit: www.arduino.cc/en/Main/Software

Arduino tarkvaraga (IDE) kirjutatud programm on *sketch*. *Sketch* koostatakse tekstiredaktoris ja salvestatakse .ino faililaiendiga. Tekstiredaktor võimaldab teksti lõigata/kleepida ja otsida/asendada. Teavituste väli alumises osas (Joonis A4) annab tagasisidet salvestamise ja eksportimise ajal ning kuvab veateateid. Valitud arendusplaat ja jadaliidese port on kuvatud akna alumises paremas nurgas. Tööriistariba nupud võimaldavad teil kontrollida ja laadida üles programme, luua, avada ja salvestada *sketch* faile ning näha jadaliidese liiklust (et näha, milliseid andmeid andurid on salvestanud). Joonis A4 kujutab kuvatõmmist Arduino IDE graafilise kasutajaliidese.



↑ Arduino IDE kasutajaliides

Arduino käsud: Tabel A1

	Verify (kontrolli). Kontrollib, kas koodis esineb vigu kompileerimise käigus.
	Upload (lae üles). Kompileerib koodi ja laeb selle seadistatud arendusplaadile.
	New (uus). Loob uue <i>sketch</i> faili.
	Open (ava). Avab menüü, kus on esitatud kõik <i>sketch</i> failid.
	Save (salvesta). Salvestab <i>sketch</i> faili.
	Serial Monitor (jadaliidese liikluse jälgimine). Avab jadaliidese liikluse jälgija, et oleks võimalik näha andurite salvestatud andmeid.

Täiendavad käsud on leitavad viie menüü alt: File, Edit, Sketch, Tools ja Help.

→ Tegevus 1: Pane mind vilkuma!

Järgnevas tegevuses vilgutavad õpilased Arduinoga valgusdiodi. Eesmärgiks on panna valgusdiod vilkuma ja mõista, kuidas kirjutada/kasutada koodi, millega igat valgusdiodi vilgutada. Rohkem teavet leiab, kui sisestada Google'i otsingumootoris: *Arduino LED blink*.

Seadmed

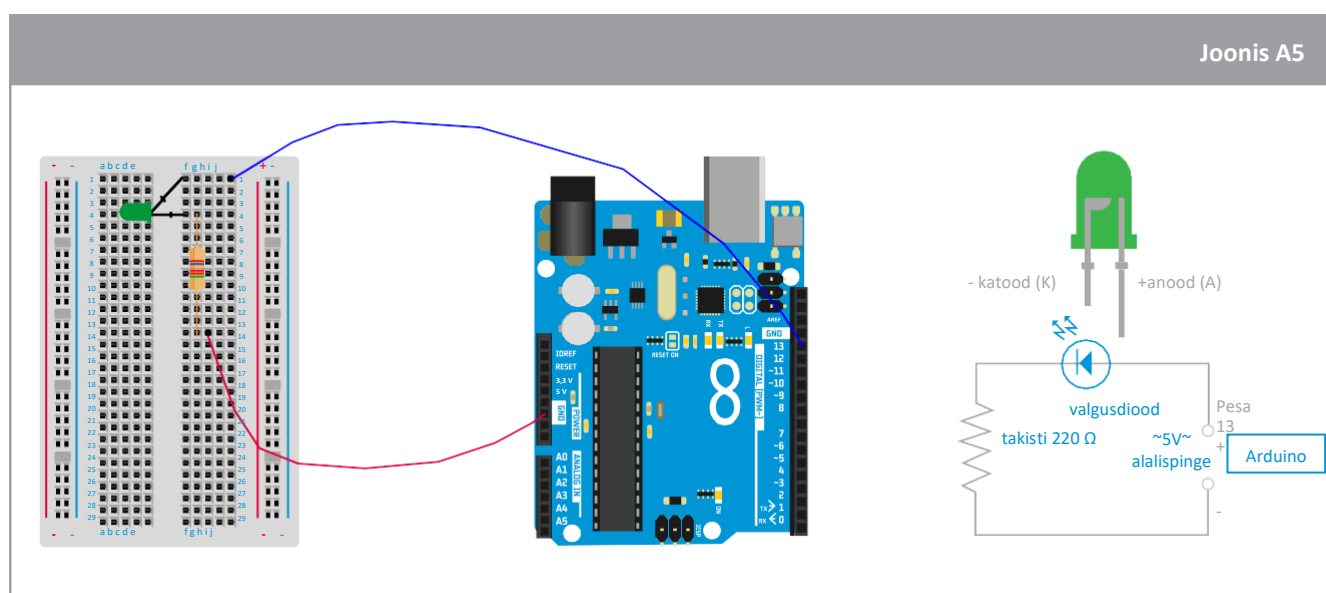
- 1 Arduino Uno
- 1 makettplaat
- 1 valgusdiod (roheline)
- 1 220 Ω takisti (punane/punane/must/must/pruun)
- 2 juhett

Ülesanne

Seadistus

Vooluring seadistatakse järgnevalt:

- Ühenda roheline valgusdiod makettplaadile nii nagu kujutatud Joonisel A5.
- Ühenda 220 Ω väärtusega takisti (punane/punane/must/must/pruun) valgusdiodi lühema jalaga nii nagu kujutatud Joonisel 5.
- Ühenda valgusdiodi pikem jalg juhtmega Arduino Uno arendusplaadi pesasse numbriga **13**.
- Ühenda takisti juhtmega Arduino Uno arendusplaadi pesasse tähisega **GND**.
- Lõpuks ühenda Arduino UNO arvutiga USB-kaabli abil.



↑ Arduino vooluring on valmis rohelise valgusdiodi vilgutamiseks

Kood

Õpilased laevad üles programmi, mille abil valgusdiodi vilgutada. Nad kasutavad näidiskoodi, mis on leitav käivitades Arduino tarkvara (IDE) ja valides:

File → Examples → Basics → Blink

Blink kood kuvatakse ekraanil ning see on esitatud järgneval joonisel. Kommentaarid järgnevad kaldkriipsudele ja on hallis kirjas ning selgitavad igat koodirida.

Programmi üleslaadimine Arduinole

```

1  /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
7  of use the correct LED pin whatever is the board used.
8  If you want to know what pin the on-board LED is connected to on your Arduino model, check
9  the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
10
11  This example code is in the public domain.
12
13  modified 8 May 2014
14  by Scott Fitzgerald
15
16  modified 2 Sep 2016
17  by Arturo Guadalupi
18  */
19
20
21 // the setup function runs once when you press reset or power the board
22 void setup() {
23   // initialize digital pin LED_BUILTIN as an output.
24   pinMode(LED_BUILTIN, OUTPUT);
25 }
26
27 // the loop function runs over and over again forever
28 void loop() {
29   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
30   delay(1000); // wait for a second
31   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
32   delay(1000); // wait for a second
33 }
34
35

```

Hall tekst tähistab kommentaare, mis selgitavad, mida kood teeb

LED_BUILTIN on pesa nimi/number, kuhu valgusdiod on ühendatud

1000 ms on viivitus enne valgusdiodi sisselülitamist

HIGH = valgusdiod on sisse lülitatud
LOW = valgusdiod on välja lülitatud

Allpool on selgitatud ülal esitatud koodi põhielemendid:

```
void setup()
```

Käesolev käsk kirjutatakse alati programmi algusesse. See lähtestab kõik muutujad, mida programmis kasutama hakatakse. Seda käsku täidetakse ainult ühe korra, et Arduino arendusplaati sisse lülitada või lähtestada.

```
pinMode(LED_BUILTIN, OUTPUT) -> pinMode(13, OUTPUT)
```

Muudab valitud pesa sisendiks või väljundiks.

Asenda vilgutamise (Blink) koodis **LED_BUILTIN** selle pesa numbriga, millega valgusdiod on ühendatud ehk numbriga **13**.

```
void loop()
```

Käesolev käsk teeb täpselt seda, millele selle nimi viitab ja jooksub järjest kõik käsked, mis selle käsu sisse on kirjutatud, võimaldades sinu programmil järjest mitut asja teha.

```
digitalWrite(13, HIGH)
```

Lülitab valgusdiodi sisse. KÕRGE (HIGH) tõstab pinget 5 voldini pesas 13.

```
delay(1000)
```

Peatab programmi määratud ajaks (millisekundites).

```
digitalWrite(13, LOW)
```

Lülitab valgusdiodi välja muutes pinget MADALAKS (LOW). Madal tähendab, et pinget pesas 13 on 0 volti.

Kõik Arduino käsud leiad, kui sisestad Google'i otsingumootoris: *Arduino commands*.

→ Tegevus 2: SOS-i vilgutamine

Järgnevas tegevuses kasutavad õpilased Arduino vooluringi, mille nad tegevuses 1 ehitasid. Nad vilgutavad rohelist valgusdiodi, et edastada teade morse koodi abil, et näidata, kuidas oleks võimalik Marsil kulguriga suhelda. Selleks, et edastada morsega „O“ täht, tuleb koodis muuta viivituse (*delay*) parameetri väärtust, et roheline valgusdiod põleks kauem. Antud juhul sisestasime 1000 ms asemele 5000 ms, kuid ajaks võib sisestada mistahes numbrit, mis on suurem kui 1000 ms. Kood on esitatud Joonisel A6:

Joonis A6

```

void loop() {
    digitalWrite(13, HIGH);
    delay(5000);
    digitalWrite(13, LOW);
    delay(1000);

    digitalWrite(13, HIGH);
    delay(5000);
    digitalWrite(13, LOW);
    delay(1000);

    digitalWrite(13, HIGH);
    delay(5000);
    digitalWrite(13, LOW);
    delay(1000);

}

```

↑ Kood morsega „O“ tähe edastamiseks

→ Tegevus 3: Temperatuuri mõõtmine

Järgnevas tegevuses mõõdavad õpilased Arduinoga temperatuuri. Eesmärgiks on mõõta temperatuurianduriga (termistor*) temperatuuri ja mõista, kuidas kirjutada/kasutada koodi, millega andurit juhtida. Rohkem teavet saab leida, kui sisestada Google'i otsingumootoris: *Arduino thermistor sensor*.

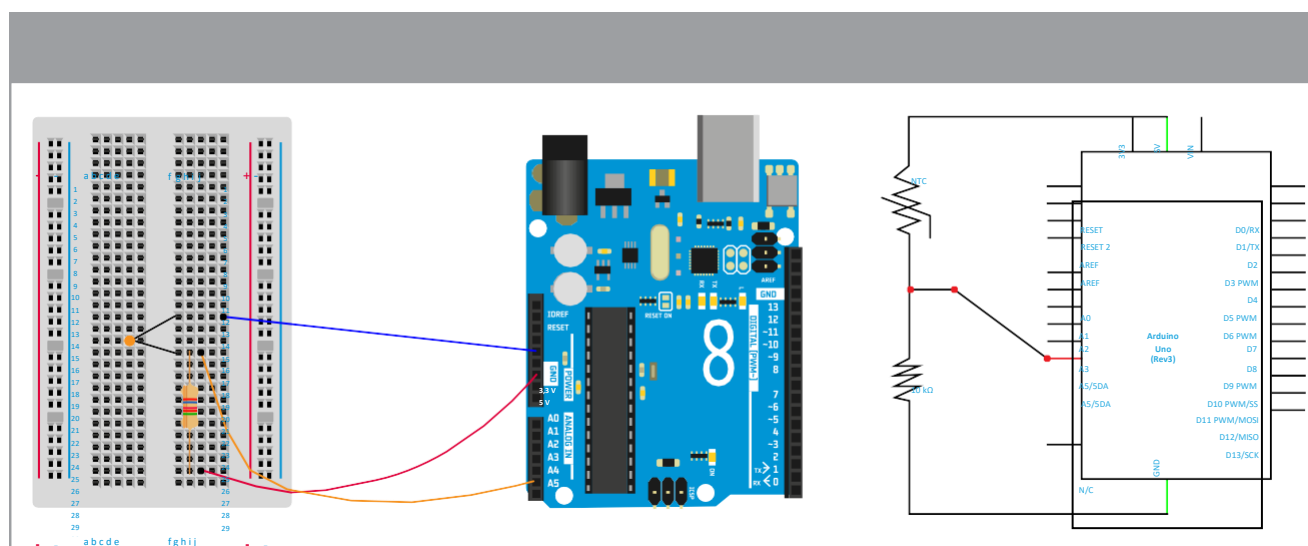
Seadmed

- 1 Arduino Uno
- 1 makettplaat
- 1 temperatuuriandur (termistor*)
- 1 10 kΩ takisti (pruun/must/must/punane/pruun)
- 3 juhett

Ülesanne

Seadistus

- Ühenda termistor makettplaadile nii nagu kujutatud Joonisel A7.
- Ühenda 10 kΩ väärtusega takisti (pruun/must/must/punane/pruun) termistoriga nii nagu kujutatud Joonisel A7.
- Ühenda termistor juhtmega Arduino Uno arendusplaadi pesasse tähisega **5V**.
- Ühenda takisti juhtmega Arduino Uno arendusplaadi pesasse tähisega **GND**.
- Ühenda termistor ja takisti juhtmega Arduino Uno arendusplaadi pesasse tähisega **A5**.
- Lõpuks ühenda Arduino UNO arvutiga USB-kaabliga.



↑ Arduino vooluring on valmis temperatuuri mõõtmiseks

* termistor: elektritakisti, mille takistus tavaliselt kuumutamise tagajärele oluliselt väheneb

Kood

Õpilased kirjutavad nüüd ise koodi. Kõik Arduino käsud leiata, kui sisestate Google otsingumootorisse: *Arduino commands*. Joonisel A8 on esitatud näidiskood temperatuuri mõõtmiseks. Käesolevas näites on anduriga mõõdetud temperatuuriks **AnalogT** ja selle ühikuks volt, kuna see on elektrisignaal; sellel pole temperatuuri füüsikalist tähendust. Temperatuur, mis on teisendatud Celsiuse kraadidesse (temperatuuriühik, mis on kasutusel Euroopas) on **AnalogTf**. Arduino Uno pesa tähiseks on A5. Kui kood on üles laetud ja käivitatud, avage jadaliidese liikluse jälgimise aken (Serial Monitor*), et näha, kuidas andmeid salvestatakse.

Joonis A8

```

1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   float AnalogT;
9   float AnalogTf;
10  AnalogT= float(analogRead(A5));
11  AnalogTf = (-40000/AnalogT) + 100;
12  Serial.println();
13  Serial.print("Temperature: ");
14  Serial.print(AnalogTf);
15  Serial.print("C");
16  delay(1000);
17
18 }
```

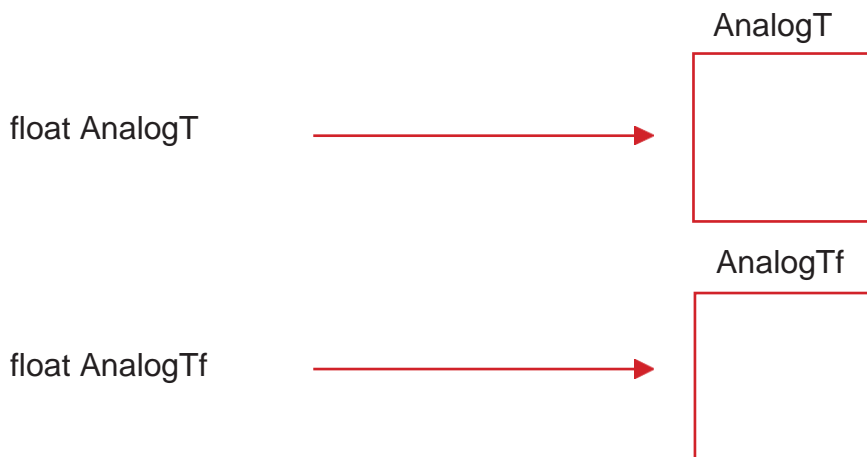
↑ [Arduino kood temperatuuri mõõtmiseks](#)

*vt Tabelit A1: *Arduino commands*.

Tegevuse 1 ülesandes 3 on kirjeldatud *void setup* ja *void loop* funktsioonid.

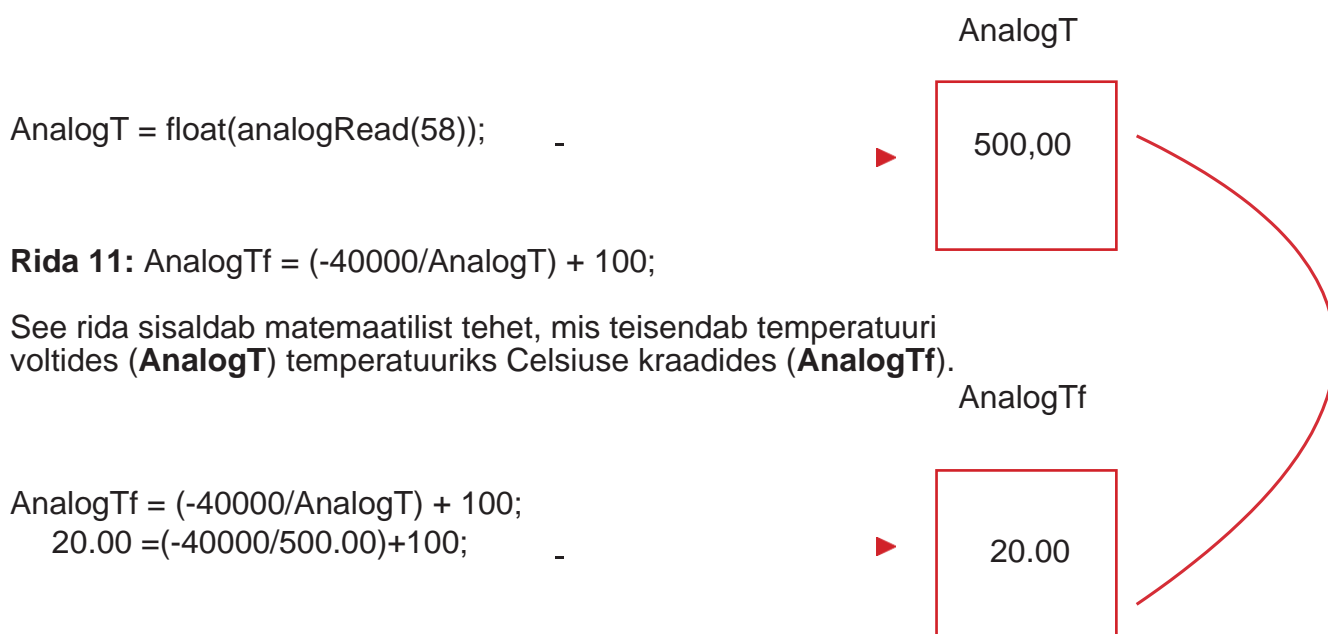
Rida 3: Temperatuuri kuvamiseks arvutiekraanil on vaja määrata andmeedastuskiirus (9600 bitti/sekundis) andmete edastamiseks Arduino ja arvuti vahel.

Read 8-9: Nendel kahel koodireal defineeritakse muutujad AnalogT ja AnalogTf. Need nimed määrab programmeerija. Selle protsessi kirjeldamiseks kasutame graafilist selgitust. Kirjutades float AnalogT, loote te arvuti mällu kasti või tühja ruumi, mille tähiseks on AnalogT ja mille suurus on määratud 'float' tüübi abil. Ujukomaarvu tüüp float viitab sellele, et kastis olev number sisaldab koma.



Rida 10: `AnalogT = float(analogRead(A5));`

Arvuti kirjutab AnalogT kasti sisse numbrit, mis loetakse Arduino Uno A5 pesast (nt 500,00). Kasutame funktsiooni **analogRead()** , et lugeda mistahes Arduino pesast.



Read 12-15: Mõõdetud temperatuuri kuvamiseks ekraanil tuleb kasutada *print* funktsiooni. See funktsioon edastab mistahes kastidest numbrid ja väljastab need Serial monitori kaudu. Käesolevas näites kuvab funktsioon numbrit AnalogTf kasti sees. Kõik sõnad, mis on esitatud “...” sees kirjutatakse otse jadaliidese liikluse jälgimise aknasse.

Rida 16: Viivituse (*delay*) funktsioon teeb programmis 1000 ms pikkuse pausi, et temperatuuri jadaliidese liikluse jälgimise aknas kuvada.

Arutelu

Õpilased peavad tõenäoliselt oma temperatuuriandurid ära kalibreerima. Nad võivad kasutada etalonina teist termomeetrit, mille abil tegelikku temperatuuri anduri kalibreerimiseks mõõta. Anduri kalibreerimiseks käivitatakse esialgne kood ja kui õpilaste temperatuuriandmed ei vasta etaloni temperatuurile, võivad nad viimast numbrit (+ 100) koodi 11. real (AnalogTf) muuta Joonisel A8.

Praktiline ülesanne

Õpilased peavad kalibreerima temperatuurianduri selliselt, et seda saaks kasutada Marsi pinnal. Marsi pinna keskmine temperatuur on ligikaudu -60 °C, Maal on see +14 °C, õpilased peavad nihke suuruseks võtma -74. Koodi tuleb kohandada järgnevalt:

$$\text{AnalogTf} = (-40000/\text{AnalogT}) + 100 - 74;$$

→ Tegevus 4: Rõhu mõõtmine

Järgnevas tegevuses mõõdavad õpilased Arduinoga ümbritsevat rõhku. Eesmärgiks on mõõta anduriga (MPX4115A) rõhku ja mõista, kuidas kirjutada/kasutada koodi, millega andurit juhtida. Rohkem teavet leiate, kui sisestate Google otsingumootoris: Arduino MPX4115A.

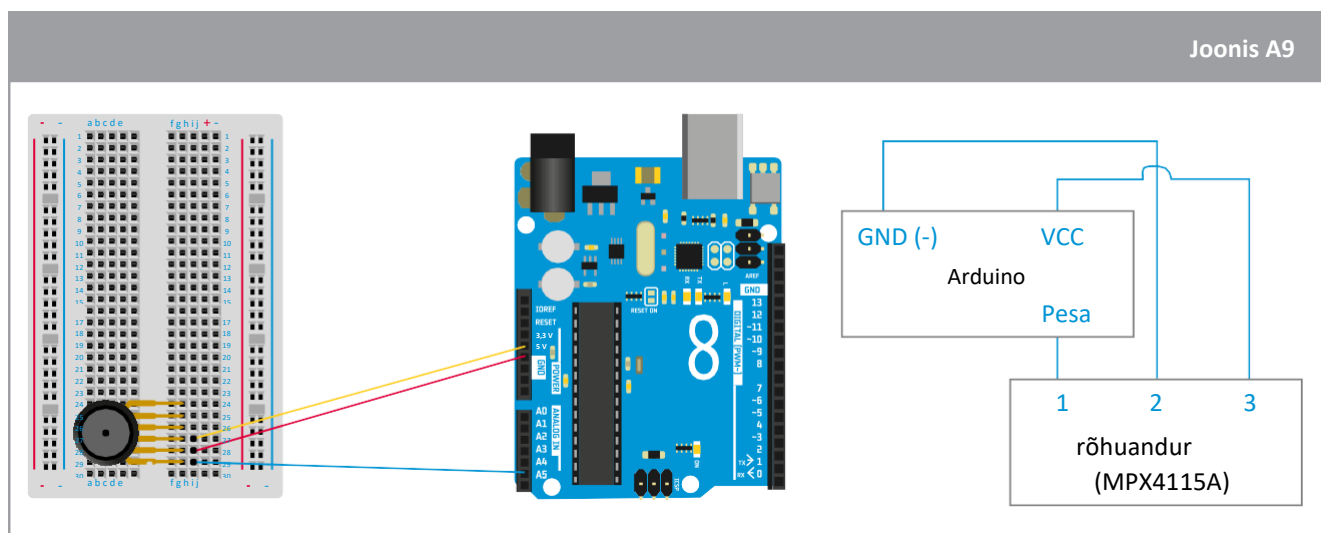
Seadmed

- 1 Arduino Uno
- 1 makettplaat
- 1 rõhuandur
- 3 juhet

Ülesanne

Seadistus

- Ühendage rõhuandur makettplaadile.
- Ühendage rõhuanduri esimene jalg (tähistatud sälguga) pesasse numbriga **A5**.
- Ühendage rõhuanduri teine jalg Arduino pesasse **GND**.
- Ühendage rõhuanduri kolmas jalg Arduino pesasse **5V**.
- Lõpuks ühendage Arduino UNO arvutiga USB-kaabliga.



↑ Arduino vooluring on valmis rõhu mõõtmiseks

Kood

Antud juhul kasutavad õpilased samu käske nagu eelneva temperatuuri tegevuse puhul; keskendutakse anduri kalibreerimisele. Andurite kasutamisel on üheks oluliseks ja väga kasulikuks sammuks nende andmelehtede lugemine, kus on detailne teave selle kohta, kuidas andur töötab. Saame lihtsa vaevaga leida anduri kohta kasulikke teavet Google otsingumootorist: *MPX4115A datasheet*.

Rõhuandur teisendab mõõdetud rõhu pingeks. See on analoogsignaali, mis teisendatakse digitaalsignaali. Mõõdetud pinget teisendamisel ümbritsevaks rõhuks (Pa või paskalitesse rahvusvahelises mõõtühikute süsteemis) on vaja kasutada anduri ülekandefunktsiooni. See funktsioon kirjeldab matemaatilist seost anduri väljundpinge ja vastava rõhu vahel. Funktsioon on leitav anduri andmelehelt. Funktsioon võib erinevate andurite puhul erineda.

Andmelehelt leiame järgmise ülekandefunktsiooni: $V_{out} = V_s (P \times 0.009 - 0.095)$. Peame ainult P (rõhu) väärtuse eraldama ja võtma arvesse, et $V_s = 1024$ (loetud pinget on jagatud 1024 sammuga). V_{out} = Analoogväärtus, mille loeme pesast **A5**.

Järgnevalt on esitatud näidiskood, mis väljastab rõhu mõõtetulemused. Kasutage seda viitena.

Joonis A10

```

1 void setup() {
2   // put your setup code here, to run once:
3   Serial.begin(9600);
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8   float AnalogP;
9   float AnalogPf;
10  AnalogP= float(analogRead(A5));
11  AnalogPf = ((AnalogP/1024)+0.095)/0.009;
12  Serial.println();
13  Serial.print("Pressure: ");
14  Serial.print(AnalogPf);
15  Serial.print("kPa");
16  delay(1000);
17
18 }
```

[↑ Arduino kood rõhu mõõtmiseks](#)

Rõhu väärtust voltides tähistatakse AnalogP ja rõhu väärtust paskalites tähistatakse AnalogPf. Pesa, kuhu rõhuandur on ühendatud, on A9.

Rõhuanduri kalibreerimiseks võivad õpilased internetist uurida, kui suur õhurõhk nende asukohas on. Anduri kalibreerimiseks käivitatakse esialgne kood ja kui õpilaste rõhuandmed ei vasta võrdlusrõhule, võivad nad koodi 11. rida (**AnalogPf**) vastavalt kohandada.

→ Tegevus 5: Kõrguse mõõtmine

Kõrgusega rõhk muutub, seega võib kõrguse mõõtmiseks kasutada rõhu väärtust. Kõrguse mõõtmiseks peavad õpilased lisama selle osa koodist rõhu mõõtmise tsükli sisse.

Järgnevalt on esitatud näidiskood, mis väljastab kõrguse meetrites. Kasutage seda viitena.

Joonis A11

```
float Altitude;
float Altitudef;
  Altitude = pow(AnalogPf/101.325,0.1903);
  Altitudef = (1-Altitude)*44300 + 100;
Serial.print("Altitude: ");
Serial.print(Altitudef); Serial.print ("meters");
```

[↑ Arduino kood rõhu mõõtmiseks](#)

Kõrgus meetrites on tähistatud muutujaga Altitude. Selles näites on rõhk kilopaskalites tähistatud muutujaga AnalogPf.

Tänu vihjetele, mis on esitatud „Kas sa teadsid?“ peatükis, võiksid õpilased mõista, et kõrguse kasvades atmosfääri temperatuur langeb ja see suhe peaks olema kõrguse ja temperatuuri graafikul näha.

Kuidas Arduinot koolis kasutada

Arduino on suurepärase seade, mille abil õpilastele programmeerimist õpetada ning tutvustada neile erinevaid rakendusi, mida seade võimaldab. Seda saab ka kasutada selleks, et õpetada ja õppida STEM aineid. Mõned näited:

- **Algteadmised voluringidest** – Tegevust 1 võib kasutada selleks, et õpilastele lihtsat voluringi õpetada.
- **Sissejuhatus andmeanalüüsi** – Õpilased koguvad andmeid pärismaailmast (temperatuur ja/või rõhk) ja analüüsivad andmeid graafikute abil. Õpilased saavad uusi teadmisi ja teevad järeldusi.
- **Ilmaprognoosi algteadmised ja andmete kogumine** – Õpilased võivad ehitada enda ilmajaama ja ennustada päikeselisi/pilviseid päevi.

→ TUTVUME ARDUINOGA!

Sissejuhatus Arduino programmeerimisse C++ keeles

Kas sa teadsid?



ExoMarsi programmi missioon 2020 viis Marsi pinnale Euroopa kulguri ja Vene teadusinstrumentid, et selgitada välja, kas Marsil on kunagi eksisteerinud elu!

→ Tegevus 0: Alustamine

Me kõik kasutame iga päev tehnoloogiat! Kosmosemissioonides kasutavad insenerid tehnoloogiat, et robotitele käsked edastada ja nendega suhelda, et avardada oma teadmisi. Järgnevas tegevuses olete teie insener, kes uurib Marssi ehk punast planeeti, kasutades Arduino nimelist tehnoloogiat. Uurime, milliseid komponente me Arduinoga kasutama hakkame!

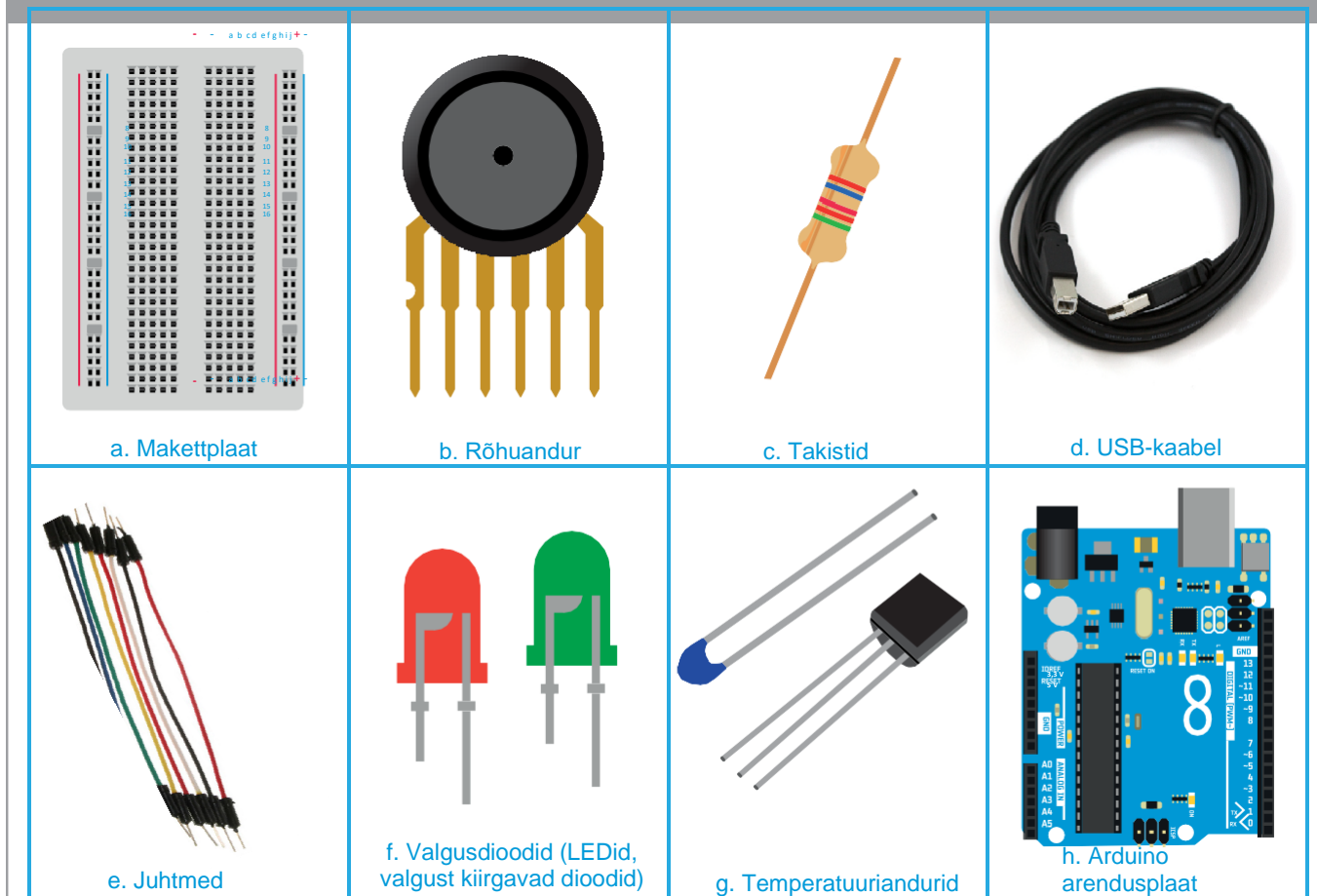
Seadmed

- Arduino põhikomponendid

Maajaama komponendid:

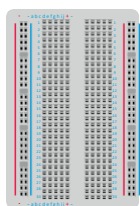
1. Paarides töötamine. Võtke oma Arduino komplekt lahti ja sobitage komponendid järgmisel leheküljel olevate kirjeldustega.

Joonis A1

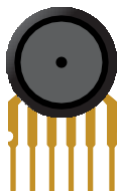


↑ Arduino põhikomponendid

KOMPONENDID



Makettplaat



Rõhuandur



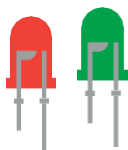
Takistid



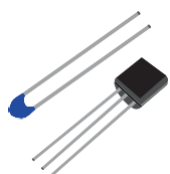
USB-kaabel



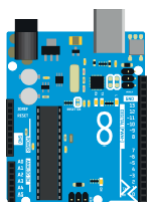
Juhtmed



Valgusdiodid (LEDid, valgust kiirgavad diodid)



Temperatuuriandurid



Arduino arendusplaat

KIRJELDUSED

on ümara kujuga ja
mõõdab
ümbritsevat
õhurõhku

on sisend- ja
väljundpesadega ja
töötas nagu lihtne arvuti

on valge tugiplaat ja
võimaldab elektroonika-
komponente lihtsalt
ühendada

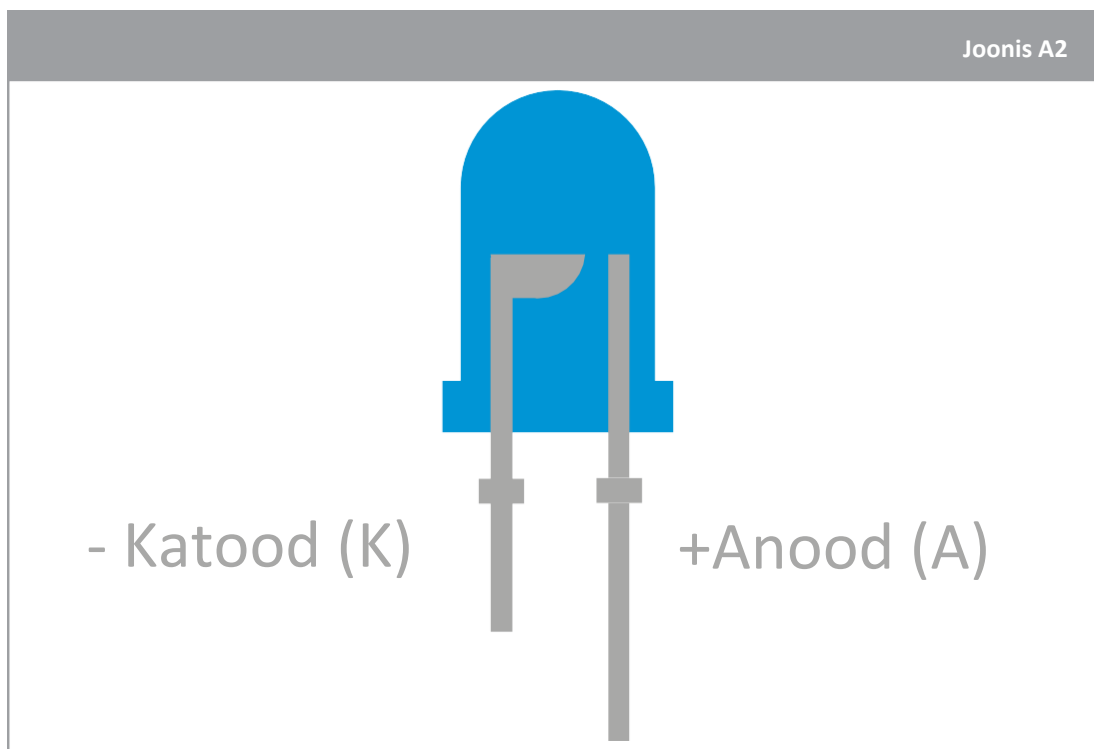
on kahe pika jalaga, mis
on sinise korpuse küljes
ja mõõdab
temperatuuri

silindrikujuline värviliste
triipudega, kasutusel
vooluringis elektrivoolu
vähendamiseks

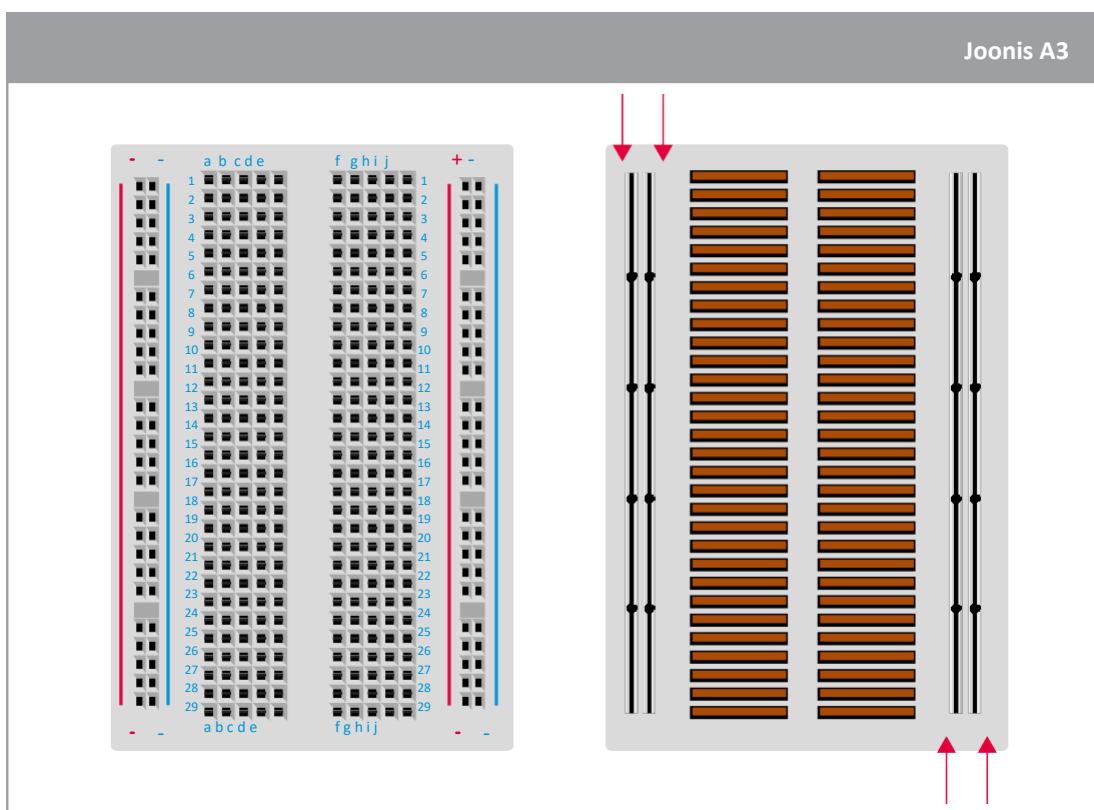
on punased ja rohelised ja
kiirgavad valgust, kui vool
neid läbib

on erinevare värvidega
ja kasutatakse selleks, et
komponente ühendada
ja elektrivoolu juhtida

pikk must kaabel,
millega ühendatakse
Arduino arvutiga



↑ Valgusdiod: Pange tähele, et valgusdiodil on üks pikk ja üks lühike jalg. Pikka jalga kutsutakse anoodiks ja lühikest katoodiks.



↑ Makettplaat: Võtke teadmiseks, et iga rea kõik pesad on omavahel ühendatud.

→ Tegevus 1: Pane mind vilkuma!

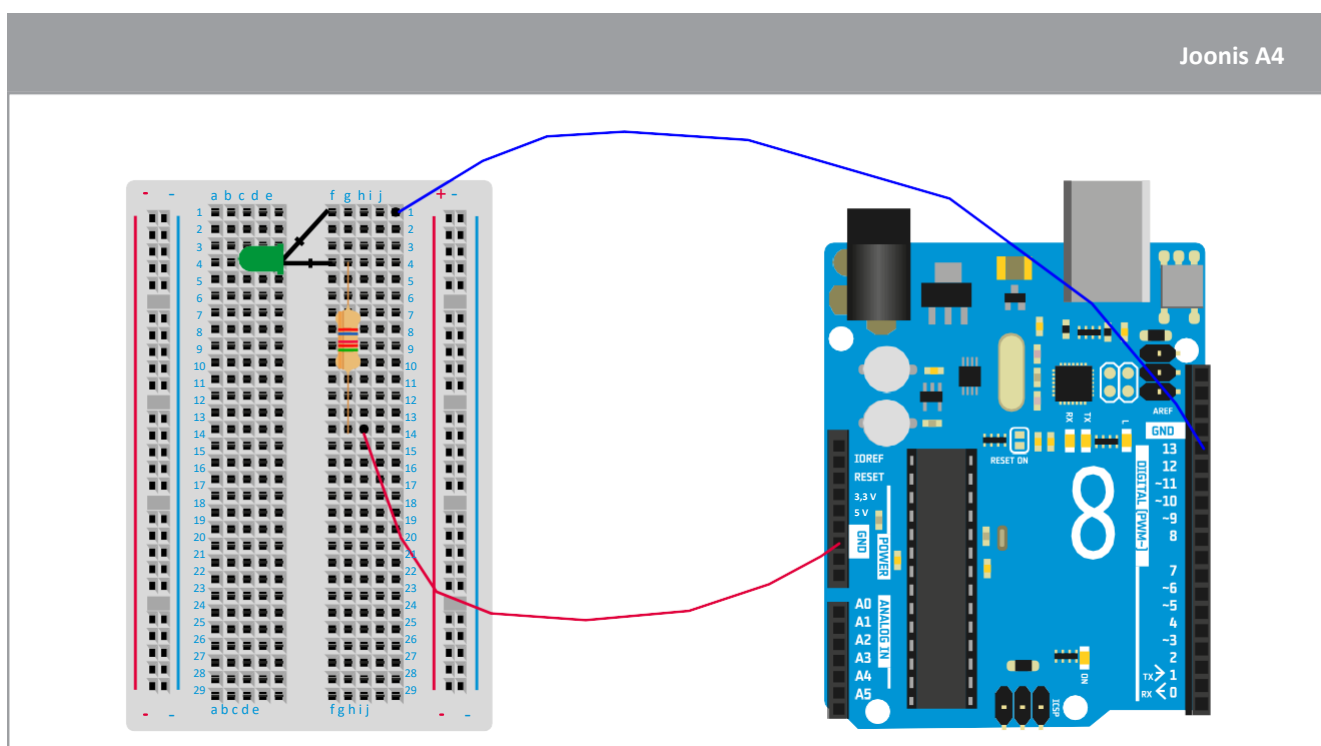
Interaktiivsete seadmete juhtimiseks on vaja kasutada tööriista nagu Arduino, mis on üles ehitatud erinevatest elektroonikakomponentidest. Kuna te olete insenerid, kelle ülesandeks on punase planeedi uurimine, peate suhtlema kulguriga, et edastada teaduseksperimentide läbiviimiseks käsk ja koguda andmeid. Järgnevas tegevuses õpite C++ programmeerimiskeelt kasutama, mis võimaldab teil Arduino Unoga suhelda ja te õpite valgusdiodi vilgutama.

Seadmed

- 1 Arduino Uno
- 1 makettplaat
- 1 roheline valgusdiod
- 1 220 Ω takisti (punane/punane/must/must/pruun)
- 2 juhet

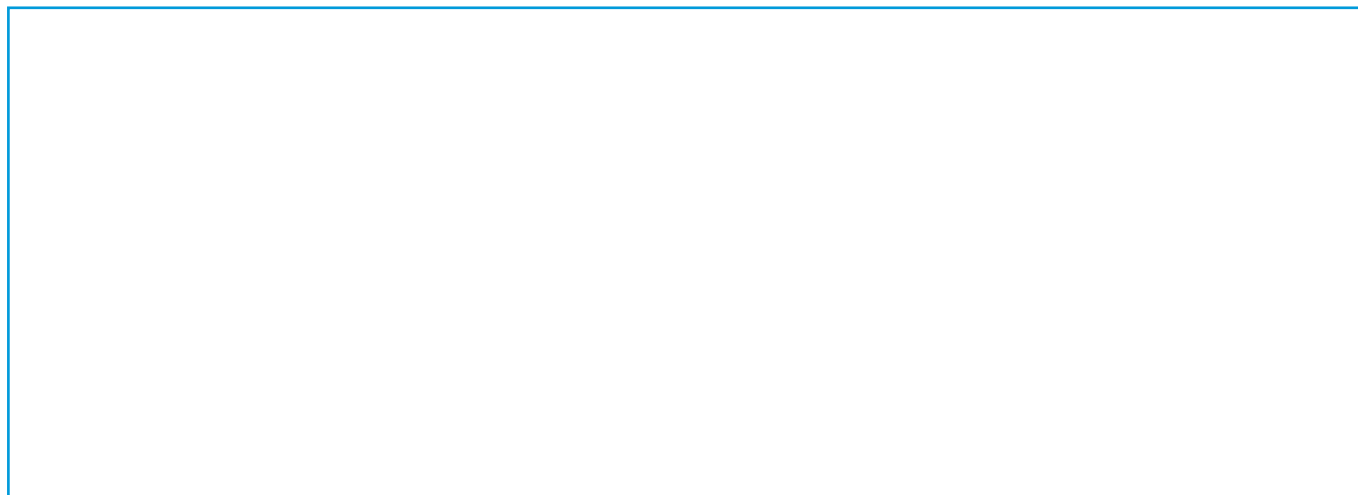
Ülesanne

- Järgige põhjalikult järgmiseid juhiseid, et makettplaat üles seada ja roheline valgusdiod vilkuma panna:
 - Ühendage roheline valgusdiod (läbipaistev plastikust diod) makettplaadile nii nagu kujutatud Joonisel A4.
 - Ühendage 220 Ω väärtusega takisti (punane/punane/must/must/pruun) valgusdiodi lühema jalaga nii nagu kujutatud Joonisel A4.
 - Ühendage valgusdiodi pikem jalg juhtmega Arduino Uno arendusplaadi pesasse numbriga **13**.
 - Ühendage takisti juhtmega Arduino Uno arendusplaadi pesasse tähisega **GND** (maandus).
 - Ühendage Arduino UNO arvutiga USB-kaabli abil.



↑ Arduino vooluring on valmis rohelse valgusdiodi vilgutamiseks

2. Joonistage allpool olevasse kasti voluring, mis vastab Joonise A4 elektrilistele ühendustele. Kasutage järgmisi sümboleid:



3. Nüüd, kui makettplaadi ühendused on valmis, peate saatma Arduino Unole valgusdiodi vilgutamiseks juhised.

Käivitage Arduino tarkvara (IDE) oma arvutis ja valige File → Examples → Basics → Blink.

Blink kood kuvatakse ekraanil ning see on esitatud järgneval joonisel. Lugege põhjalikult selgitust, et kõigest täielikult aru saada:

Programmi üleslaadimine Arduinole

```

1  /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
7  of use the correct LED pin whatever is the board used.
8  If you want to know what pin the on-board LED is connected to on your Arduino model, check
9  the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
10
11 This example code is in the public domain.
12
13 modified 8 May 2014
14 by Scott Fitzgerald
15
16 modified 2 Sep 2016
17 by Arturo Guadalupi
18 */
19
20
21 // the setup function runs once when you press reset or power the board
22 void setup() {
23   // initialize digital pin LED_BUILTIN as an output.
24   pinMode(LED_BUILTIN, OUTPUT);
25 }
26
27 // the loop function runs over and over again forever
28 void loop() {
29   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
30   delay(1000); // wait for a second
31   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
32   delay(1000); // wait for a second
33 }
34
35

```

Hall tekst tähistab kommentaare, mis selgitavad, mida kood teeb

LED_BUILTIN on pesa nimi/number, kuhu valgusdiod on ühendatud

HIGH = valgusdiod on sisse lülitatud

LOW = valgusdiod on välja lülitatud

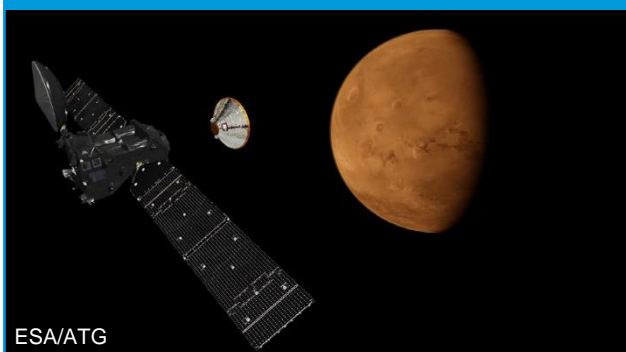
1000 ms on viivitus enne valgusdiodi sisselülitamist

- a. Enne koodi üleslaadimist Arduino Unole peate veenduma, et pesa tähis koodis vastab pesa tähisele, kuhu roheline valgusdiod Arduino Unol on ühendatud. Kirjeldage, mida te muutsite oma rohelise valgusdiodi vilgutamise koodis:

- b. Laadige kood Arduino Unole, vajutades vasakus nurgas asuvale  nupule.

Õnnitleme, roheline valgusdiod vilgub!

Kas sa teadsid?



Marss on nii kaugel Maast, et signaalide edastamine kosmoselaevast või kulgurist tagasi Maale võtab kaua aega. See viivitus signaalide edastamisel muudab kulguriga suhtlemise ja kiire reageerimise ootamatutele olukordadele suureks väljakutseks.

→ Tegevus 2: SOS-i vilgutamine

Kuna te oskate nüüd valgusdiodi Arduino koodi abil vilgutada, õpetab järgnev tegevus teile, kuidas suhelda valgusdiodi abil ja edastada teadet Marsil olevale kulgurile. Tegelikult on tugev liivatorm kulguri suunas liikumas! Selleks, et vältida kulguri kahjustamist, peate te kulguri SOS teadet edastama panema!

Kas sa teadsid?



Sellel pildil on näha liivatormi Marsil. Nähtavad pilved on tekkinud atmosfäärirõhu, temperatuuri ja kõrguse muutumise tagajärjel vertikaalse nihke tõttu, mis esineb näiteks siis, kui tuul puhub üle mäe või kraatri seina.

Seadmed

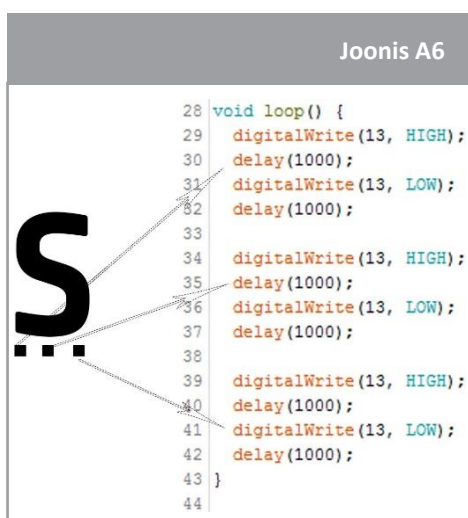
- Vooluring, mille ehitasite tegevuses 1

Ülesanne

Kasutage morse koodi, et roheline valgusdiode vilkuma panna. Joonisel A5 tähistab punkt lühikest signaali, mis paneb valgusdiodi korraks põlema ja joon pikka signaali, mis paneb valgusdiodi põlema pikemaks ajaks. Kood morsega „S“ tähe edastamiseks on näitena esitatud Joonisel A6.



↑ SOS morse koodiga



↑ Kood morsega „S“ tähe edastamiseks

Kirjeldage, mida te koodis muutama peate, et edastada morsega „O“ täht:

Kohandage koodi selliselt, et see edastaks „SOS“ teate, seejärel laadige see Arduinole ja katsetage seda!

→ Tegevus 3: Temperatuuri mõõtmine

Arduino tehnoloogia võimaldab teil vilgutada sülearvutis kirjutatud koodi abil valgusdiodi, mis on ühendatud Arduino Unoga. Kui te laete koodi üles, edastab see Arduino Unole juhised ja lülitab elektroonikakomponendid sisse. ExoMarsi missioonis kasutatakse tehnoloogiat selleks, et andurite abil Marsi keskkonda uurida. Järgnevas tegevuses õpite klassiruumi temperatuuri mõõtma ja simuleerite temperatuuri planeedil Marss!

Seadmed

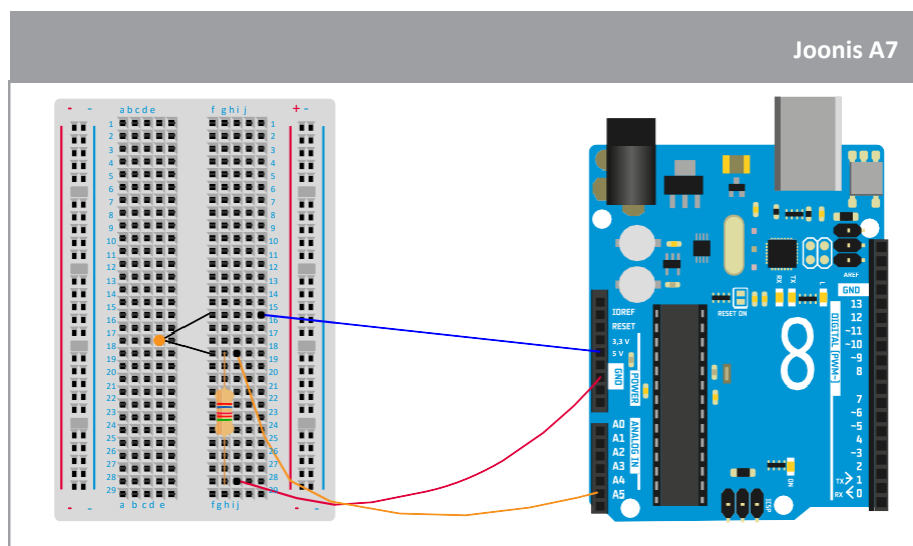
- 1 Arduino Uno
- 1 makettplaat
- 1 temperatuuriandur (termistor)
- 1 10 k Ω takisti (pruun/must/must/punane/pruun)
- 3 juhet

Ülesanne

1. Klassiruumi temperatuuri mõõtmiseks peate te ehitama makettplaadile voluringi nagu järgnevalt on kujutatud.

Järgige hoolikalt juhiseid:

- a. Ühendage termistor makettplaadile nii nagu kujutatud Joonisel A7.
- b. Ühendage 10 k Ω väärtusega takisti (pruun/must/must/punane/pruun) termistoriga nii nagu kujutatud Joonisel A7.
- c. Ühendage termistor juhtmega Arduino Uno arendusplaadi pesasse tähistega **5V**.
- d. Ühendage takisti juhtmega Arduino Uno arendusplaadi pesasse tähistega **GND**.
- e. Ühendage termistor ja takisti juhtmega Arduino Uno arendusplaadi pesasse tähistega **A5**.
- f. Ühendage Arduino UNO arvutiga USB-kaabliga.



↑ Arduino voluring on valmis temperatuuri mõõtmiseks

2. Joonisel A8 on kujutatud osaliselt poolik kood temperatuuri mõõtmiseks. Tuleb sisestada kolm väärtust, mis on esitatud kastikes (punane, roheline ja sinine), et Arduino Unole temperatuuri mõõtmiseks õiged juhised edastada. Uurime, kuidas seda koodi täiendada.

```

Joonis A8
1
2 void setup() {
3   Serial.begin(9600);
4 }
5
6 void loop() {
7
8   float ;
9   float ;
10
11    = float(analogRead());
12    = (-40000/)+100;
13
14   Serial.println();
15   Serial.print("Temperature in classroom: ");
16   Serial.print();
17   Serial.print("C");
18   delay(1000);
19
20 }
```

↑ [Arduino kood temperatuuri mõõtmiseks](#)

a. Mis ühikuga Euroopas temperatuuri tähistatakse?

Rõhuanur, mis on Arduino Unoga ühendatud, mõõdab kõigepealt temperatuuri teises ühikus ehk voltides. Selleks, et teisendada see number voltidest Celsiuse kraadidesse, kasutate valemit, mis on esitatud 12. real Joonisel A8. Lõpuks kuvatakse temperatuur ekraanil kasutades koodis *Serial.print* funktsiooni.

b. Defineerige muutuja, mis tähistab temperatuuri voltides:

Defineerige muutuja, mis tähistab temperatuuri Celsiuse kraadides:

Otsige üles Arduino pesa, mille külge temperatuuriandur on ühendatud:

c. Te olete valmis selleks, et ise oma sülearvutis kood kirjutada. Käivitage Arduino tarkvara ja sisestage Joonisel A8 esitatud kood koos muutujatega, mis te olete defineerinud punase, rohelise ja sinise kasti jaoks. Asendage hoolikalt iga värviline kast vastava väärtusega!

Laadige kood Arduino Unole vajutades nupule.

d. Temperatuuri mõõtetulemuste nägemiseks tuleb Arduino tarkvaras avada eraldi aken, mille nimi on Serial monitor. Otsige selle avamiseks tööriistaribalt järgmist sümbolit



Arutelu

1. Arutage temperatuuri mõõtmise asjakohasust oma kaaslastega. Kuidas on võimalik veenduda selles, et temperatuuriandur mõõdab klassiruumi temperatuuri täpselt?

2. Millist rida Joonisel A8 esitatud koodist tuleb kohandada, et koodi abil oleks võimalik andurit kalibreerida?

3. Selleks, et Arduino kasutamist veel rohkem harjutada, kalibreerige temperatuuriandur selliselt, et seda saaks kasutada Marsi pinnal. Arvestage, et Marsi pinna keskmine temperatuur on ligikaudu $-60\text{ }^{\circ}\text{C}$.

a. Maa keskmine temperatuur =

b. Sisestage temperatuurianduri uus kalibreerimise valem siia:

→ Tegevus 4: Rõhu mõõtmine

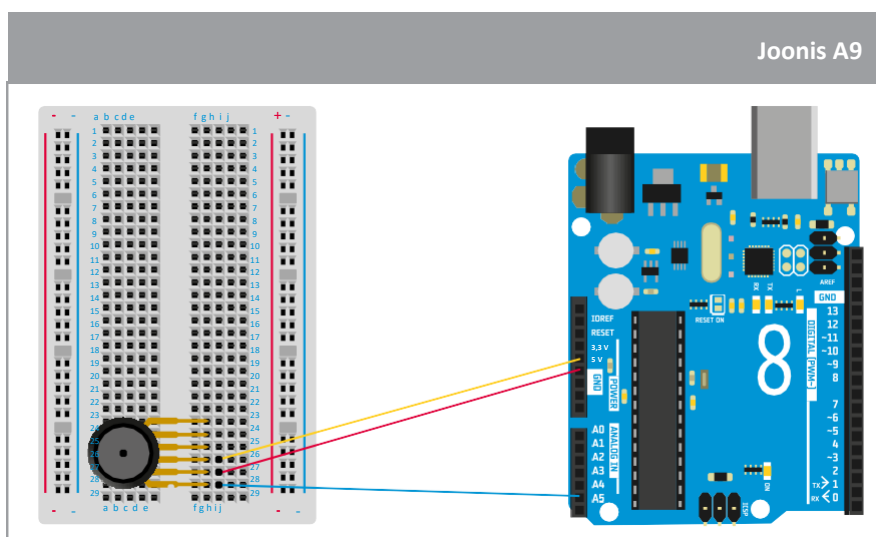
Järgnevas tegevuses õpite Arduino ja rõhuanduriga mõõtma atmosfäärirõhku.

Seadmed

- 1 Arduino Uno
- 1 makettplaat
- 1 rõhuandur
- 3 juhet

Ülesanne

1. Klassiruumi rõhu mõõtmiseks peate te ehitama makettplaadile voluringi (Joonis A9). Järgige hoolikalt järgmiseid juhiseid (märkus: iga rõhuanduri jalg käitub erinevalt, seega peab andur olema samamoodi asetatud nagu joonisel):



↑ Arduino voluring on valmis rõhu mõõtmiseks

2. Joonisel A10 on kujutatud osaliselt poolik kood rõhu mõõtmiseks. Tuleb sisestada kolm muutujat, mis on esitatud kastikes (punane, roheline ja sinine), et Arduino Unole rõhu mõõtmiseks õiged juhised edastada. Uurime, kuidas seda koodi täiendada.

```

Joonis A10

1 void setup() {
2
3   Serial.begin(9600);
4 }
5 void loop() {
6
7   float [red box] ;
8   float [green box] ;
9
10  [red box] = float(analogRead([blue box]));
11  [green box] = ([red box] / 1024) + 0.095) / 0.009;
12  Serial.println();
13  Serial.print("Pressure in classroom: ");
14  Serial.print([green box]);
15  Serial.print("kPa");
16  delay(1000);
17
18 }
```

- a. Millise füüsilise ühikuga määratakse rõhk rahvusvahelises mõõtühikute süsteemis?

- b. Defineerige muutuja, mis tähistab rõhku voltides:

Defineerige muutuja, mis tähistab rõhku kilopaskalites:

Otsige üles Arduino pesa, mille külge rõhuandur on ühendatud:

Rõhuandur, mis on Arduino Unoga ühendatud, mõõdab kõigepealt rõhku teises ühikus, voltides, mis on elektripingega ühikuks. Selle numbri teisendamiseks voltidest paskalitesse kasutame valemit, mille leiame anduri andmelehel $V_{out} = V_s (P \times 0.009 - 0.095)$.

Lõpuks kuvatakse rõhk ekraanil kasutades koodis Serial.print funktsiooni.

- c. Kasutage valemit $V_{out} = V_s (P \times 0.009 - 0.095)$ ja arvutage kalibreerimise valem Joonisel A10 esitatud 12. rea jaoks. Kuna loetud pinge on jagatud 1024 sammuga, on $V_s = 1024$.

- d. Te olete valmis selleks, et ise oma sülearvutis kood kirjutada. Käivitage Arduino tarkvara ja sisestage Joonisel A10 esitatud kood koos muutujatega, mis te olete defineerinud punase, roheline ja sinise kasti jaoks. Asendage hoolikalt iga värviline kast vastava väärtusega!

Laadige kood Arduino Unole vajutades  nupule

- e. Rõhu mõõtetulemuste nägemiseks tuleb Arduino tarkvaras avada eraldi aken, mille nimi on Serial monitor. Otsige selle avamiseks tööriistaribalt järgmist sümbolit: 

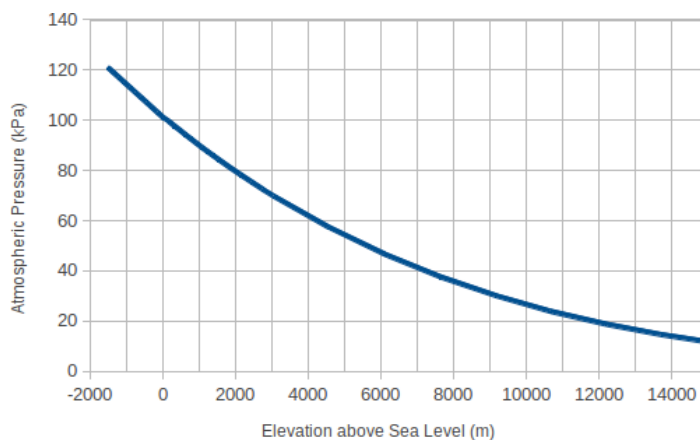
Arutelu

Arutage rõhu mõõtmise asjakohasust oma eakaaslastega. Kuidas on võimalik veenduda selles, et rõhuandur mõõdab klassiruumi rõhku täpselt?

Millist rida Joonisel A10 esitatud koodist tuleb kohandada, et koodi abil oleks võimalik andurit kalibreerida?

→ Tegevus 5: Kõrguse mõõtmine

Õhurõhk sõltub kõrgusest. Näiteks on õhurõhk merel kõrgem kui õhurõhk mägedes. Te näete graafikult, kuidas see muutub. Matemaatika ja mingil määral gaaside seaduste mõistmise abil saame välja arvutada rõhu ja kõrguse suhte. Praegu ei pea me sellepärast muretsema. Järgnevas tegevuses kasutate rõhu mõõtetulemusi selleks, et määrata oma kõrgus, kasutades lihtsustatud valemit. See kirjeldab rõhu ja kõrguse suhet, mida tuntakse baromeetrilise valemina.



Seadmed

- Arduino voluring, mille ehitasite tegevuses 4
- Kood rõhu mõõtmiseks, mida kasutasite tegevuses 4

Ülesanne

1. Avage Arduino tarkvaras rõhu mõõtmise kood, mille kirjutasite tegevuses 4. Lisage see osaliselt puuduv kood koos *void loop* funktsiooni sisse:

```

Joonis A11
1 float [ ] ;
2 float [ ] ;
3
4 [ ] = pow( [ ] / 101.325, 0.1903);
5 [ ] = (1 - [ ]) * 44300 + 100;
6
7 Serial.println();
8 Serial.print("Altitude in classroom: ");
9 Serial.print([ ]);
10 Serial.print("meters");
11 delay(1000);

```

↑ Arduino kood kõrguse mõõtmiseks

2. Kõrguse arvutamiseks kasutate rõhuandurilt saadud mõõtetulemusi valemis, mis on esitatud 4. real Joonisel A11. Nagu te eelnevates ülesannetes nägite, edastab Arduino Unoga ühendatud andur kõigepealt mõõtetulemuse voltides. Selleks, et teisendada see number voltidest meetritesse, kasutate valemit, mis on esitatud 5. real Joonisel A11. Lõpuks kuvatakse kõrguse andmed ekraanil, kasutades koodis Serial.print funktsiooni. Kui mõõtetulemus ei ole kalibreeritud, saab seda kohandada, muutes koodi 5. real.

3. a. Defineerige muutuja, mis tähistab kõrgust voltides:

Defineerige muutuja, mis tähistab kõrgust meetrites:

Otsige muutuja nime, mille andsite rohelisele kastile, kuhu salvestatakse rõhk paskalites (vaadake tegevust 4):

b. Täiustage koodi muutujatega, mis te määrasite kollasele, lillale ja rohelisele kastile. Asendage hoolikalt iga värviline kast vastava väärtusega! Seejärel laadige koos üles.

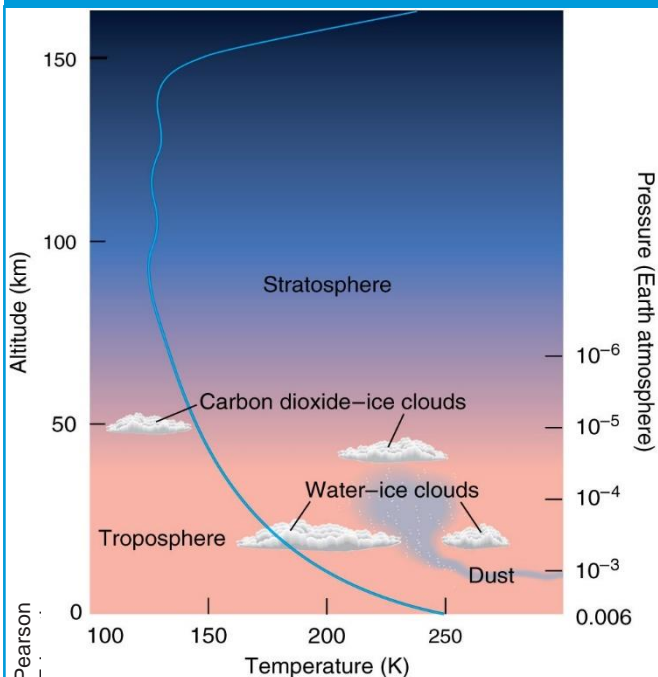
c. Kõrguse mõõtetulemuste nägemiseks tuleb Arduino tarkvaras avada eraldi aken, mille nimi on Serial monitor.

4. Kopeerige temperatuuri ja kõrguse mõõtmise koodid ühte programmi ja laadige see Arduino Unole.

Mõõtke oma klassiruumi temperatuuri ja rõhku põrandast laeni ning joonistage graafik, kus te tulemused esitate.

Kas te oskate graafiku kuju selgitada?

Kas sa teadsid?



Marsi atmosfäär koosneb õhukestest gaasikihtidest, peamiselt süsihappegaasist, mis ulatub Marsi pinnast kosmose piirini. Päike kuumutab Marsi pinda ja osa kuumusest läheb selle peale, et soojendada gaasi, mis asub Marsi pinna lähedal. Kuumutatud gaas hajub või tõuseb konvektsiooni teel läbi atmosfääri. Seega on gaasi temperatuur kõige kõrgem pinna lähedal ja kõrguse kasvades see väheneb.

→ LINGID

Arduino blogi, Arduino kohta uusima teabe leidmiseks:

<https://blog.arduino.cc/>

Juhendid erinevate Arduino komponentide kasutamiseks

<https://quarkstream.wordpress.com/>

Paljud Arduinol põhinevad projektid, mida saate kodus proovida, on leitavad Instructablesi veebileheküljelt: <http://www.instructables.com/howto/arduino/>

Palju muid Arduino projekte:

<https://hackaday.io/list/3611-arduino-projects>