

TALLINNA POLÜTEHNIKUM

IT ja telekommunikastiooni erialaosakond

Arti Hunt

ARDUINO LABORITÖÖDE JUHENDID

Lõputöö

ARVUTID JA ARVUTIVÕRGUD

IT-14E

Juhendaja: Meelis Antoi

Tallinn 2017

AUTORIDEKLARATSIOON

Deklareerin, et käesolev lõputöö, mis on minu iseseisva töö tulemus, on esitatud Tallinna Polütehnikumi lõputunnistuse taotlemiseks Arvutid ja arvutivõrgud erialal. Lõputöö alusel ei ole varem eriala lõputunnistust taotletud.

Arti Hunt
(allkiri ja kuupäev)

Töö vastab kehtivatele nõuetele

Juhendaja: Meelis Antoi
(allkiri ja kuupäev)

Sisukord

SISUKORD.....	3
1. LÜHENDITE JA MÕISTETE SÕNASTIK.....	5
SISSEJUHATUS	7
2. MIKROARVUTI PLATVORMI VALIK	9
3. ARDUINO UNO TUTVUSTUS.....	10
4. ARDUINO IDE TUTVUSTUS	12
5. PRAKTILISED TÖÖD.....	14
5.1 Ülesanne 1. Arduino tarkvara installeerimine.....	14
5.2 Digitaalsed I/O	15
5.2.1 Ülesanne 2. Nupule vajutades süttib LED	15
5.2.2 Ülesanne 3. Nupule vajutades süttib LED 1 sekundiks	16
5.2.3 Ülesanne 4. LED süttib nupu vabastamisel	17
5.3 Analoog sisendid	17
5.3.1 4.3.1. Ülesanne 5. Potentsiomeetri max nivoo ületamisel süttib LED	17
5.3.2 Ülesanne 6. LED-i vilkumise sagedus sõltub potentsiomeetri sisendist.....	18
5.4 LCD	19
5.4.1 Ülesanne 7. LCD kasutamine	19

5.4.2	Ülesanne 8. LCD kasutamine analoogsisendiga	20
5.5	Andurid JA LCD	21
5.5.1	Ülesanne 9. Temperatuuri andur koos matemaatika teegiga	21
5.5.2	Ülesanne 10. Kaugusandurid	23
5.6	Mootor.....	23
5.6.1	Ülesanne 11. Servomootor.....	23
5.6.2	Ülesanne 12. Potentsiomeetriga juhitud servomootor	24
5.7	Kommunikatsioon	25
5.7.1	Ülesanne 13. Jadaliides	25
6.	LAHEMDUSTE TESTIMINE (ÕPPEMATERJALIDE KASUTAMINE PRAKTILISES TÖÖS).....	27
7.	KOKKUVÕTE	28
8.	KASUTATUD KIRJANDUS	29
9.	LISAD	30

1. Lühendite ja mõistete sõnastik

LED - valgusdiod

Takisti - Takisti on elektroonikakomponent mingi soovitava või kindla elektritakistuse tekitamiseks vooluringis.

LCD - Vedelkristallkuvar ehk LCD-kuvar (ingl Liquid Crystal Display) on kuvar, mille talitus põhineb vedelkristallides ilmnevatel elektrooptilistel nähtustel.

USB – Infotehnoloogias on USB järjestiksiinil töötav standardiseeritud port, tänu millele saab arvuti külge ühendada välisseadmeid

Potentsiomeeter – Potentsiomeeter on muuttakisti, mida kasutatakse peamiselt elektroonikalülitustes reguleeritava pingejagurina

Surunupplüli – Lüli

Termotakisti – Termistor ehk termotakisti on termoelektriline pooljuhtseadis, mille takistus sõltub tugevalt ja mittelineaarselt temperatuurist.

Mikrokontroller - Mikrokontroller (algse nimega ühekiibi mikroarvuti) on mikrokiip, milles on peale keskprotsessori veel komponente, näiteks mälu.

SRAM - SRAM (Static Random Access Memory) ehk staatiline juhupöördlusega mälu on pooljuhi tüüp, kus sõna staatiline viitab sellele, et erinevalt dünaamilisest RAM-ist (DRAM), ei pea SRAM-i oma ehituse tõttu perioodiliselt uuendama.

EEPROM - EEPROM tähistab Electrically Erasable Programmable Read-Only Memory ehk Elektriliselt Eemaldatav Programmeeritav Lugemiskaitstud Mälu ja on püsimälu tüüp, mida kasutatakse arvutites ja teistes elektroonilistes seadmetes, et salvestada väikeseid andmeosakesi, mis peavad alles jääma pärast toite eemaldamist, näiteks kalibreerimistabelid või seadme konfiguratsioon.

Taktsagedus - Taktsagedus on mikrokiibi töökiirust iseloomustav suurus, täpsemalt taktgeneraatori tekitatud impulsside arv sekundis. Taktsagedust mõõdetakse hertsides.

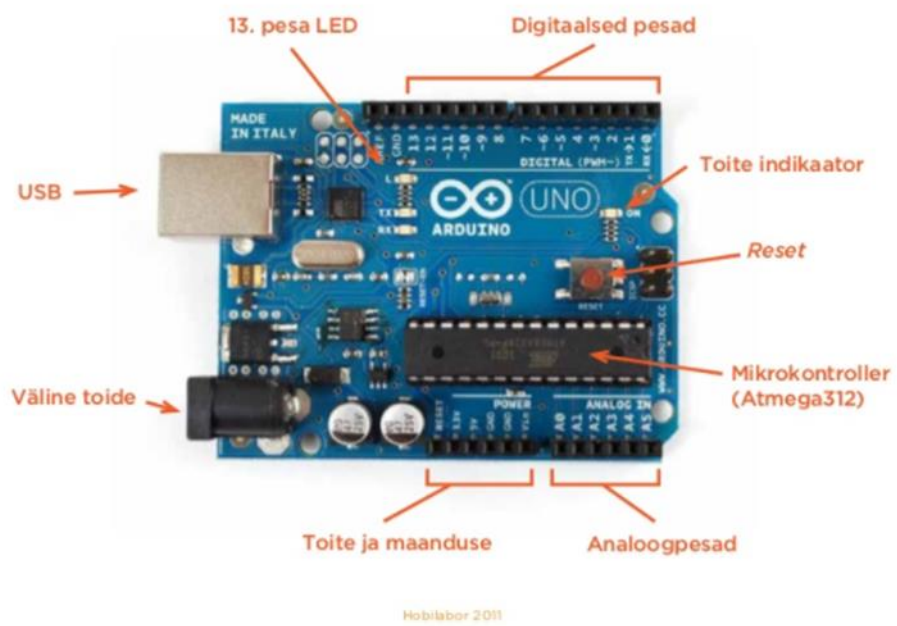
Sissejuhatus

Kooli teoreetilise töö kindlustamiseks tuleb teha saadud teadmised praktikas läbi.

Praktiste tööde edukaks läbi viimiseks on vaja koostada laborijuhendid. Et selgeks saada tegelikult arvuti toimimist, seda saab teha mikroarvutite peal. Mikroarvuti tuleks valida mis oleks odav, kõigile kättesaadav ja praktiline.

Arduino on lihtne, võimekas mikroarvuti, mis on mõeldud erinevate lahenduste automatiseerimiseks. Tänapäeval on Arduino enamlevinud platvormiks robotikaks. Selle baasil on hea õpetada mikroarvuti toimimist.

Laborijuendi ülesanded on tehtud järk-järguliselt keerukamaks, et õpetajal oleks lihtne hinnata õpilaste õpitulemusi.



Joonis 1. Ülevaade Arduino UNO sisendite ja väljuntide kohta

2. Mikroarvuti platvormi valik

Valisin platvormiks Arduino UNO, see on üks kõige populaarsem ja odavam. See on kasutusel enamustel robotikalahendustel, tekitab õpilastel palju huvi, materjalid on kergesti saadaval internetis. Arduino on hea algajatele õppimiseks mida kujutab endast programmeerimine ja robotika. Sobib ka väga hästi kooli õppekavasse. Seda on kerge kasutada, sellel on suur kogukond. Sobib erinevatele raskusastmetele, palju õppematerjale, skeemid ja lähtekoodid avatud.

Et tunnis operatiivselt töid läbi viia, oleks vaja need tööd läbi teha ja selle järgi juhend koostada. Esialgu tuleb viia läbi kergemad ülesanded ja selgitada miks ja kuidas toimib, milleks see vajalik on, ning hiljem järk-järgult teha raskemaid ülesandeid.

3. Arduino UNO tutvustus

Arduinot programmeeritakse enamasti Wiring-keelel põhineva programmeerimiskeele abil, kasutades Processing-keskkonnal põhinevat integreeritud arenduskeskkonda. Arduino programmeerimiseks kasutatav keel on sarnane C++ keelega, mõne muudatuse ja lihtsustusega.

Arduino on avatud lähtekoodiga ühe plaadi mikrokontroller, mis suudab tajuda ümbrust saades erinevatest sensoritest sisendinformatsiooni ning muuta oma ümbrust juhtides näiteks LEDe, mootoreid ja teisi ajureid. Arduino riistvara koosneb lihtsast vaba riistvara disainiga trükiplaadist, millel on Atmel AVR protsessor, ning mugavamaks kasutamiseks välja toodud erinevad sisend- ja väljundviigud. Tarkvara koosneb standardsest programmeerimiskeele kompilaatorist ja mikrokontrolleris olevast alglaadimise haldurist.

Arduino riistvarast on erinevaid versioone ja neid on võimalik osta eelnevalt kokkupandult. Riistvara on võimalik ise luua, on riistvara disaini informatsioon vabalt internetis saadaval. Lisaks Arduinole on riistvarast olemas ka teisi ühilduvuse tasemetega teiste tootjate variatsioone. Mõned neist on programmeeritavad tavapärase Arduino tarkvaraga.

Arduino projekt tõsteti esile 2006. aastal Prix Ars Electronica autasustamisel digitaalsete kogukondade kategoorias. (Arduino, 2015)

Info:

Mikrokontroller:	ATmega328
Tööpinge:	5V
Sisendpinge (soovitatav):	7-12V
Sisendpinge (piirid):	6-20V
Digitaalseid pesasid:	14 (millest 6 PWMiga)
Analoogpesasid:	6
Voolutugevus pesa kohta:	40mA
Voolutugevus 3.3V pesal:	50Ma
Mälu:	32KB (ATmega328), millest 0.5KB bootloader
SRAM:	2KB (ATmega328)
EEPROM:	1KB (ATmega328)
Taktsagedus:	16MHz
(Rene Rebane, 2011)	

4. Arduino IDE tutvustus

Arduino on open-source elektroonika prototüüpimise platvorm, mis põhineb kergestikasutataval riistvaral ja tarkvaral ning on algselt loodud kunstnikele, disaineritele ja hobielektronikutele.

Arduino plaatidel on üldjuhul kasutusel AVR mikrokontroller. Programmeerimiskeskkonnaks on Arduino IDE,

Programmi aknas on meie jaoks kõige olulisem:

Kontrollib kas kood laadimata

Kompileerib koodi ja laeb selle mikrokontrollerisse

Avab serial terminali, mille kaudu saab mikrokontrolleriga töö ajal infot vahetada või programmi debugida

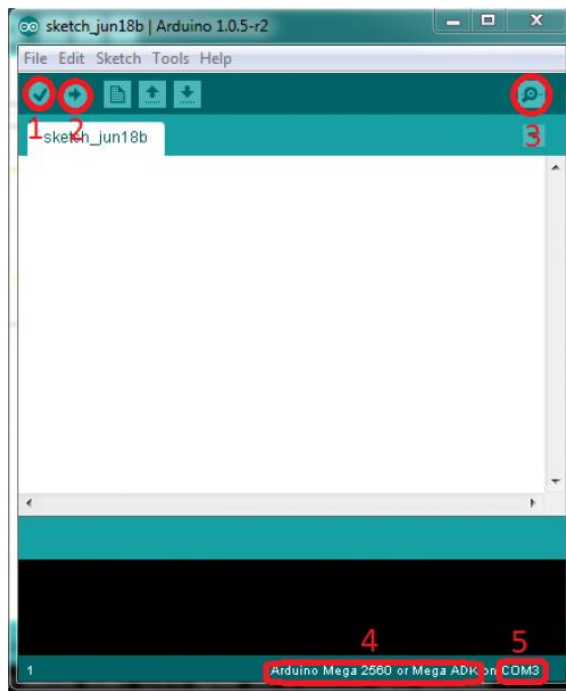
Valitud Arduino plaadi versioon

kompileerub/kas koodis on süntaksivigu, ilma programmi mikrokontrollerisse

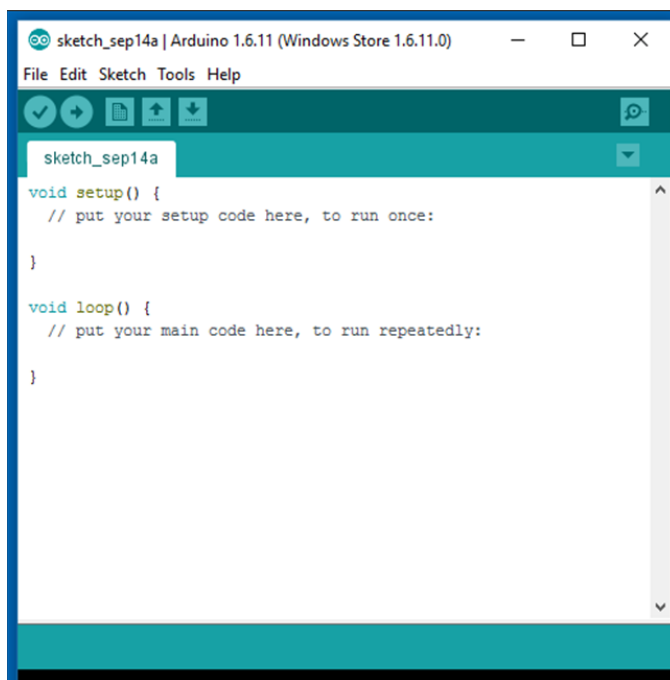
COM port, kuhu on ühendatud Arduino (vt joonis 2.)

setup() – funktsioon mis töötab korra programmi alguses ning seadistab mikrokontrolleri parameetrid.

loop() – funktsioon mis kutsutakse korduvalt esile kuni plaat välja lülitatakse.(vt joonis 3.)



Joonis 2. Ülevaade Arduino IDE tarkvarast



Joonis 3. Ülevaade Arduino IDE funktsioonide kohta

5. Praktilised tööd

5.1 Ülesanne 1. Arduino tarkvara installeerimine

Esmalt laadisime alla Arduino tarkvara leheküljelt Arduino IDE:
<https://www.arduino.cc/en/Main/Software>.

Ühendasime arduino USB-kaabilga arvutiga. Panime tööle eelnevalt alla laetud tarkvara Arduino IDE.

Mida Meil oli selleks vaja? Arvutit millel oleks vähemalt 1 vaba USB pesa, Arduino UNO ja usb-kaabel (Microsoft 2017).

5.2 Digitaalsed I/O

5.2.1 Ülesanne 2. Nupule vajutades süttib LED

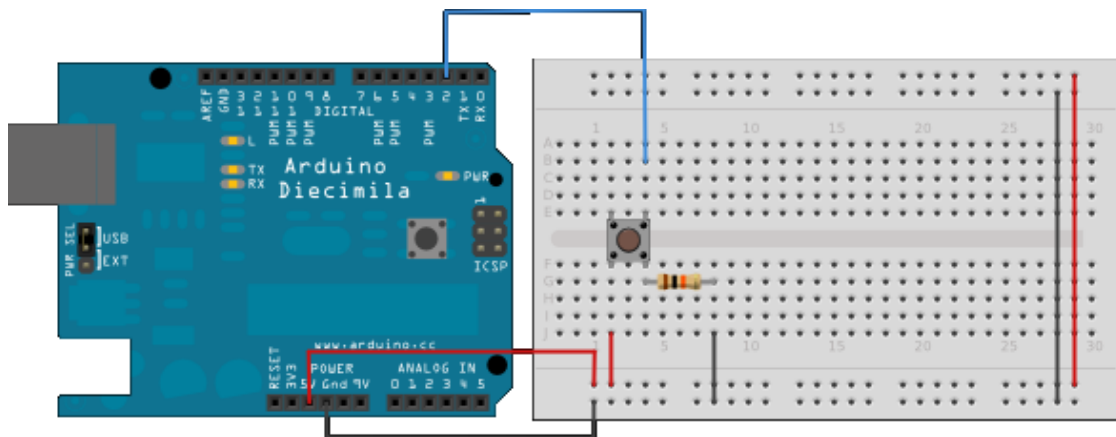
Kirjeldus: Programm demonstreerib nupu ja LED-i kasutamist.

```
const int nupp = 2; // Viik kuhu on ühendatud nupp
const int LED = 13; // Viik kuhu on ühendatud nupp roheline LED
// Globaalsed muutujad
int NupuOlek = 0; // Nupu oleku muutuja

void setup() {
  pinMode(LED, OUTPUT); // algväärtustame LED viigu väljundiks
  pinMode(nupp, INPUT); // algväärtustame nupu viigu sisendiks
  digitalWrite(nupp, HIGH); // lülitame sisemised pullup takistid sisse
}

void loop(){
  NupuOlek = digitalRead(nupp); // salvestame muutujasse nupu hetke
  väärtuse
  if (NupuOlek == LOW) { // Kui nupu on alla vajutatud
    digitalWrite(LED, HIGH); // süütame LED-i
  }
  else { // vastasel juhul
    digitalWrite(LED, LOW); // kustutame LED-i
  }
}
```

(vt joonis 4.) (Sell R., 2015)



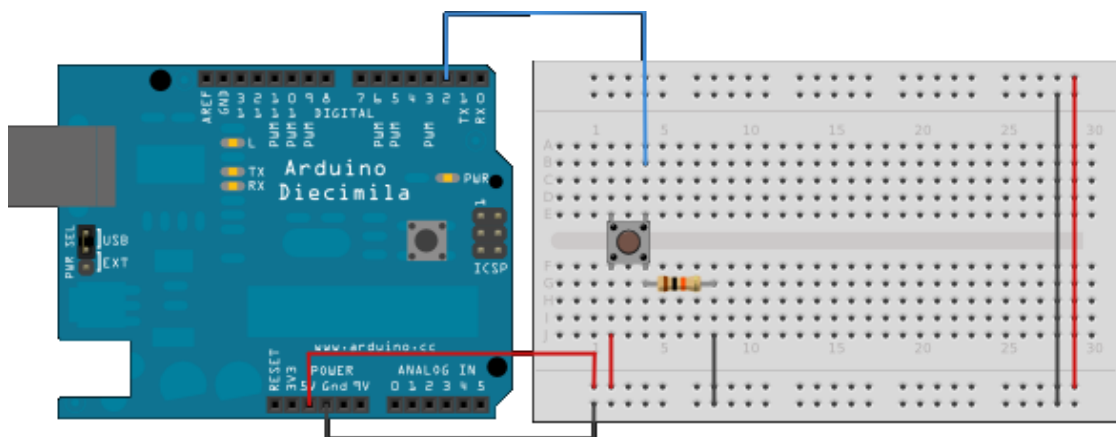
Joonis 4.

5.2.2 Ülesanne 3. Nupule vajutades süttib LED 1 sekundiks

Kirjeldus: Programm demonstreerib nupu ja LED-i kasutamist.

```
// Konstandid
const int nupp = 2; // Viik kuhu on ühendatud nupp
const int LED = 13; // Viik kuhu on ühendatud nupp roheline LED
// Globaalsed muutujad
int NupuOlek = 0; // Nupu oleku muutuja
void loop(){
    if (digitalRead(nupp) == LOW) { // Kui nupu olek on madal
        digitalWrite(LED, HIGH); // süütame LED-i
        delay(1000); // ootame 1 sekundi (1000 millisekundit)
    }
    digitalWrite(LED, LOW); // kustutame LED-i
}
```

(vt joonis 5.) (Sell R., 2015)



Joonis 5.

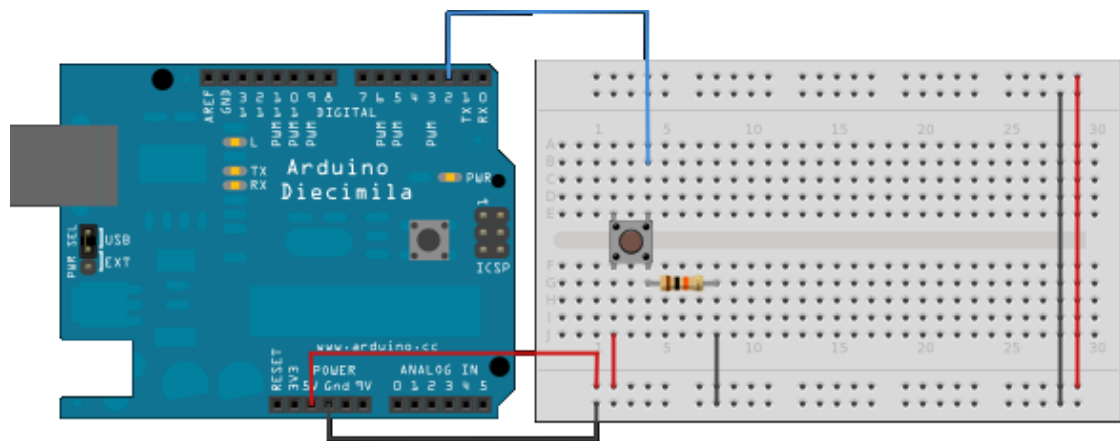
5.2.3 Ülesanne 4. LED süttib nupu vabastamisel

Kirjeldus: Programm demonstreerib nupu ja LED-i kasutamist.

```
// Konstandid
const int nupp = 2; // Viik kuhu on ühendatud nupp
const int LED = 13; // Viik kuhu on ühendatud nupp roheline LED
// Globaalsed muutujad
int NupuOlek = 0; // Nupu oleku muutuja

void loop(){
  if (digitalRead(nupp) == LOW) { // Nupule vajutus
    while (digitalRead(nupp) == LOW){} // Ootame nupu lahtilaskmist
    digitalWrite(LED, HIGH); // süütame LED-i
  }
}
```

(vt joonis 6) (Sell R., 2015)



Joonis 6.

5.3 Analoo sisendid

5.3.1 4.3.1. Ülesanne 5. Potentsiomeetri max nivoo ületamisel süttib LED

Kirjeldus: Programm demonstreerib analoogsisendi kasutamist.

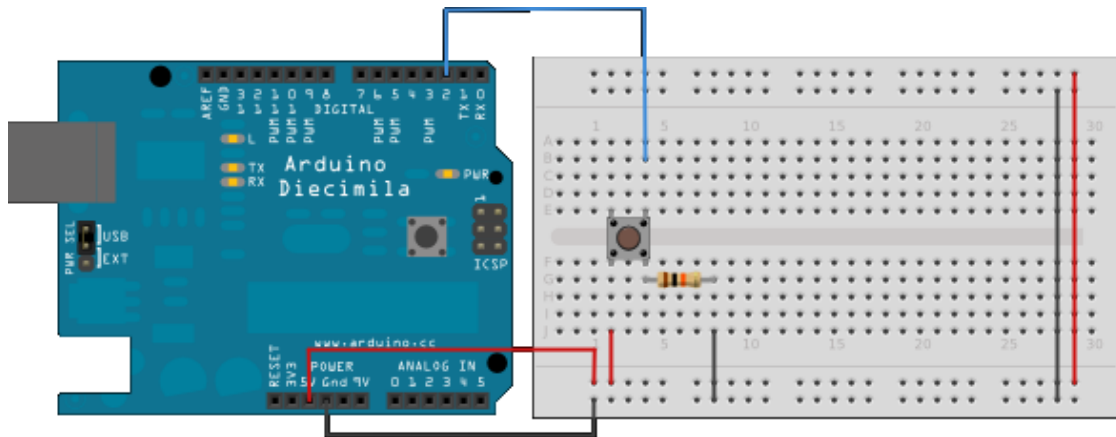
```
// määrame nivoo, millest kõrgemal väärtusel //süüdatakse LED
const int nivoo = 512;

int pote_sisend = 2; // määrame potentsiomeetri sisendviigu
int led = 13; // määrame LED ühendusviigu
int pote = 0; // muutuja potentsiomeetri väärtuse salvestamiseks

void setup() {
  pinMode(led, OUTPUT); // algväärtustame LED viigu väljundiks
}
```

```
void loop() {
  pote = analogRead(pote_sisend); // loeme analoogsisendi väärtuse
  // kui väärtus on suurem nivoo, süütame LED-i
  if (pote > nivoo) digitalWrite(led, HIGH);
  else digitalWrite(led, LOW); // vastasel korral kustutame LED-i
}
```

(vt joonis 7.) (Sell R., 2015)



Joonis 7.

5.3.2 Ülesanne 6. LED-i vilkumise sagedus sõltub potentsiomeetri sisendist

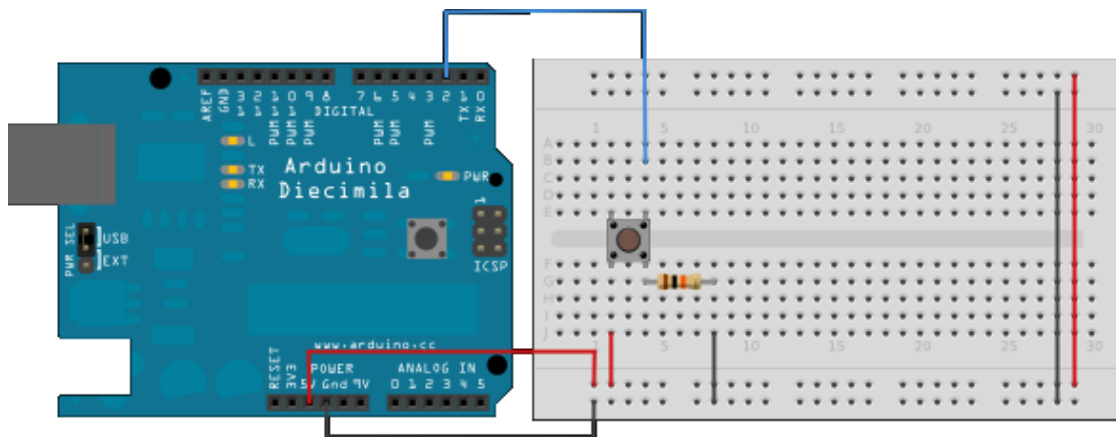
Kirjeldus: Programm demonstreerib analoogsisendi kasutamist.

```
int pote_sisend = 2; // määrame potentsiomeetri sisendviigu
int led = 13; // määrame LED ühendusviigu
int pote = 0; // muutuja potentsiomeetri väärtuse salvestamiseks

void setup() {
  pinMode(led, OUTPUT); // algväärtustame LED viigu väljundiks
}

void loop() {
  pote = analogRead(pote_sisend); // loeme anduri väärtuse
  digitalWrite(led, HIGH); // kustutame LED-i
  delay(pote); // tekitame viite <sensorValue> millisekundit
  digitalWrite(led, LOW); // süütame LED-i
  delay(pote); // tekitame viite <sensorValue> millisekundit
}
```

(vt joonis 8.) (Sell R., 2015)



Joonis 8.

5.4 LCD

5.4.1 Ülesanne 7. LCD kasutamine

Kirjeldus: Kasutab LiquidCrystal.h teeki ja ITEAD Studio 2x16 teksti laiendusplaati.

```
#include <LiquidCrystal.h> // Kaasame vajaliku teegi

// Initsialiseerime LCD koos vastavate viikude ühendamisega
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

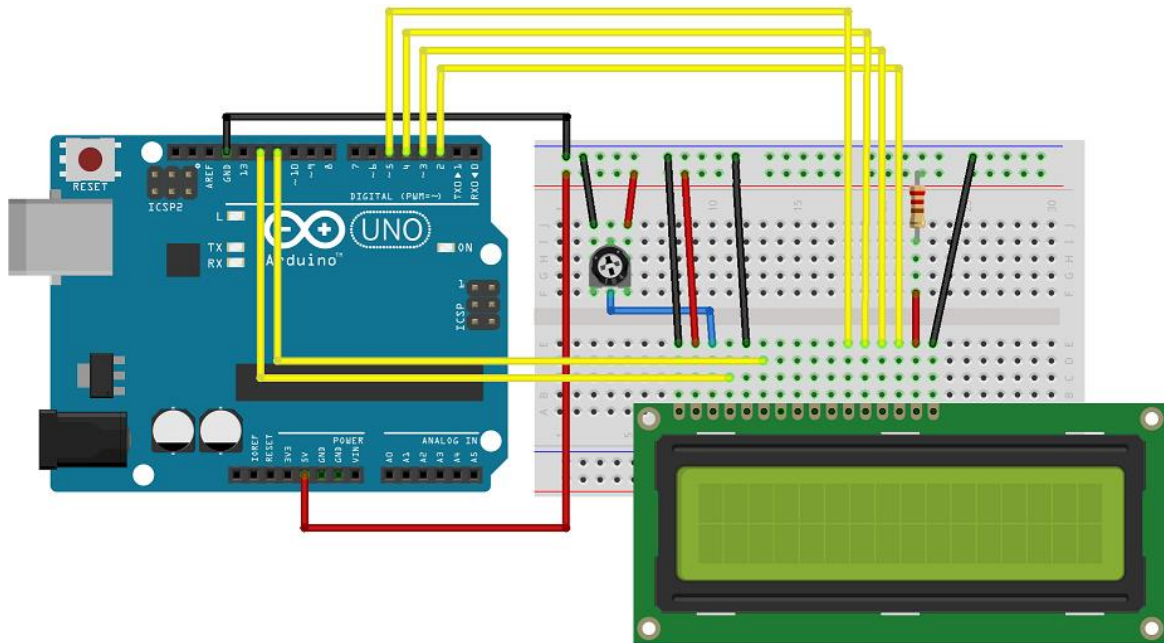
void setup() {
  lcd.begin(16, 2); //Määrame LCD read ja veerud
  lcd.print("Tere Tallinn!"); // Trükime tervitusteksti
}

void loop() {

  lcd.setCursor(0, 1); //Viime kursori esimesele reale esimesse
  positsiooni (1. rida on indeksiga 0)

  lcd.print(millis()/1000); //Trükime loenduri väärtuse
}
```

(vt joonis 9.) (Sell R., 2015)



Joonis 9.

5.4.2 Ülesanne 8. LCD kasutamine analoogsisendiga

Kirjeldus: kasutab LiquidCrystal.h teeki ja ITEAD Studio 2x16 teksti laiendusplaati.

```
#include <LiquidCrystal.h> // Kaasame vajaliku teegi

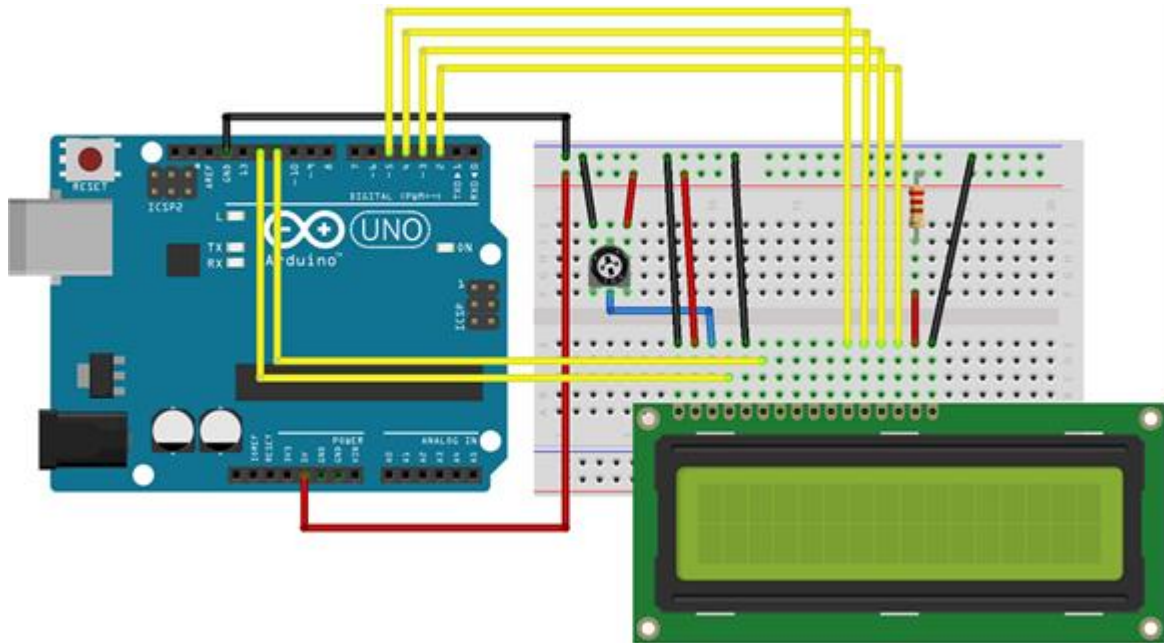
// Initsialiseerime LCD koos vastavate viikude ühendamisega
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int pote_sisend = A1; // määrame potentsiomeetri sisendviigu

void setup() {
  lcd.begin(16, 2); //Määrame LCD read ja veerud
  lcd.print("Potentsiomeeter"); // Trükime tervitusteksti
}

void loop() {

  lcd.setCursor(0, 1); //Viime kursori esimesele reale esimesse
  positsiooni (1. rida on indeksiga 0)
  lcd.print(analogRead(pote_sisend)); //Trükime analoogsisendi väärtuse
  lcd.print(" "); //Trükime tühikud, et vana väärtus ei jääks segama
  delay (100);
}
```

(vt joonis 10.) (Sell R., 2015)



Joonis 10.

5.5 Andurid JA LCD

5.5.1 Ülesanne 9. Temperatuuri andur koos matemaatika teegiga

Kirjeldus: kasutab Steinhart-Hart termistori valemit:

$$\text{temperatuur kelvinites} = 1 / \{ A + B[\ln(R)] + C[\ln(R)]^3 \}$$

kus $A = 0.001129148$, $B = 0.000234125$ ja $C = 8.76741E-08$

```
// Kaasame vajalikud teegid
#include <math.h>
#include <LiquidCrystal.h>

// Initsialiseerime LCD koos vastavate viikude ühendamisega
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
int andur = A1; // määrame temperatuurianduri (termistori) sisendviigu

void setup() {
  lcd.begin(16, 2); //Määrame LCD read ja veerud
  lcd.print("Temperatuur"); // Trükime tervitusteksti
  delay (1000);
}

void loop() {
  Termistor(analogRead(andur)); // Käivitame funktsiooni
  delay(1000); // Ootame 1 sek.
  lcd.clear(); // Puhastame ekraani vanadest andmetest
}

void Termistor(int RawADC) {
```

```

    double Temp;
    long Takistus;
    // Valem: Takistus = (1024 * JaguriTakisti/ADC) - JaguriTakisti
    Takistus=((10240000/RawADC) - 10000);
    //Esimene rida
    lcd.setCursor(0, 0);
    lcd.print("AD=");
    lcd.print(RawADC); //Trükime analoogsisendi väärtuse

    lcd.setCursor(8, 0);
    lcd.print("U=");
    lcd.print(((RawADC*5.0)/1024.0),3); //Trükime pinget väärtuse

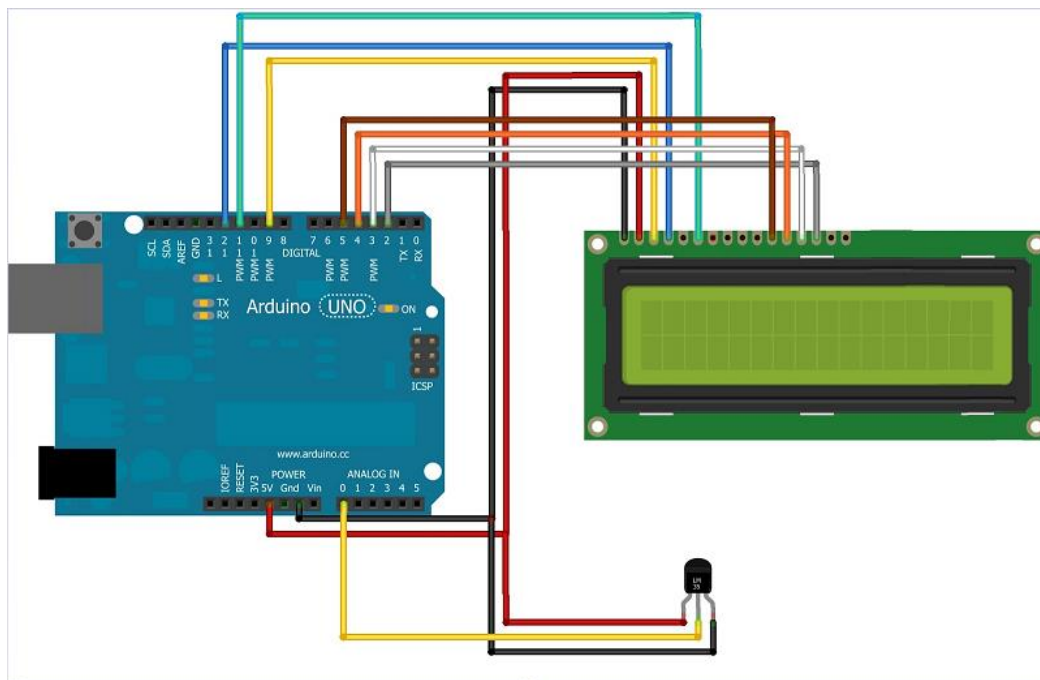
    //Teine rida
    lcd.setCursor(0, 1);
    lcd.print("R=");
    lcd.print(Takistus); //Trükime takistuse väärtuse

    Temp = log(((10240000/RawADC) - 10000));
    Temp = 1/(0.001129148 + (0.000234125 + (0.0000000876741 * Temp *
Temp)) * Temp);
    Temp = Temp - 273.15; // Konverteeri Kelvinid Celciustesse

    lcd.setCursor(8, 8);
    lcd.print("T=");
    lcd.print(Temp); //Trükime temperatuuri väärtuse
}

```

(vt joonis 11.) (Sell R., 2015)



5.5.2 Ülesanne 10. Kaugusandurid

Kirjeldus: programm demonstreerib kaugusanduri kasutamist.

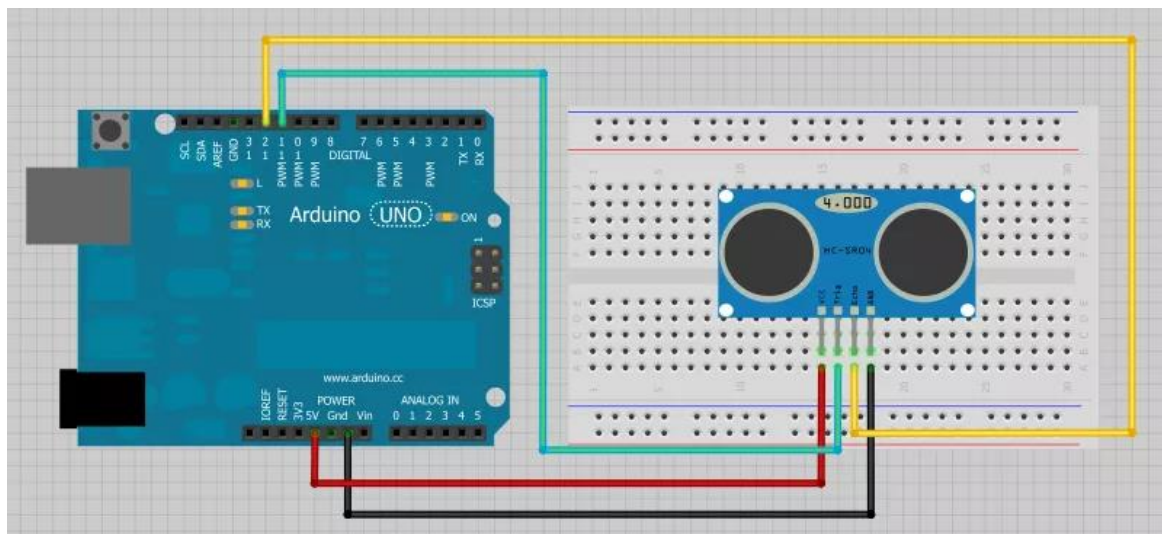
```
// Reading analog and digital sensors
void setup()
{
  // For visualization we can use serial monitor
  Serial.begin(9600);
}

void loop()
{
  // Read the input on analog pin 2 and 3:
  int AnalogSensorValue = analogRead(A2); //Sharp analog kaugusandur
  int DigitalSensorValue = digitalRead(A3); //Digitaalne lähedusandur

  Serial.println(AnalogSensorValue); // Print out the analog value
  Serial.println(DigitalSensorValue); // Print out the digital value

  delay(500); // Delay in between readings for readability
}
```

(vt joonis 12.) (Sell R., 2015)



Joonis 12.

5.6 Mootor

5.6.1 Ülesanne 11. Servomootor

Kirjeldus: programm demonstreerib servomootori kasutamist.

```
#include <Servo.h>

Servo right_motor, left_motor; // create servo object to control a servo

void setup()
{
```

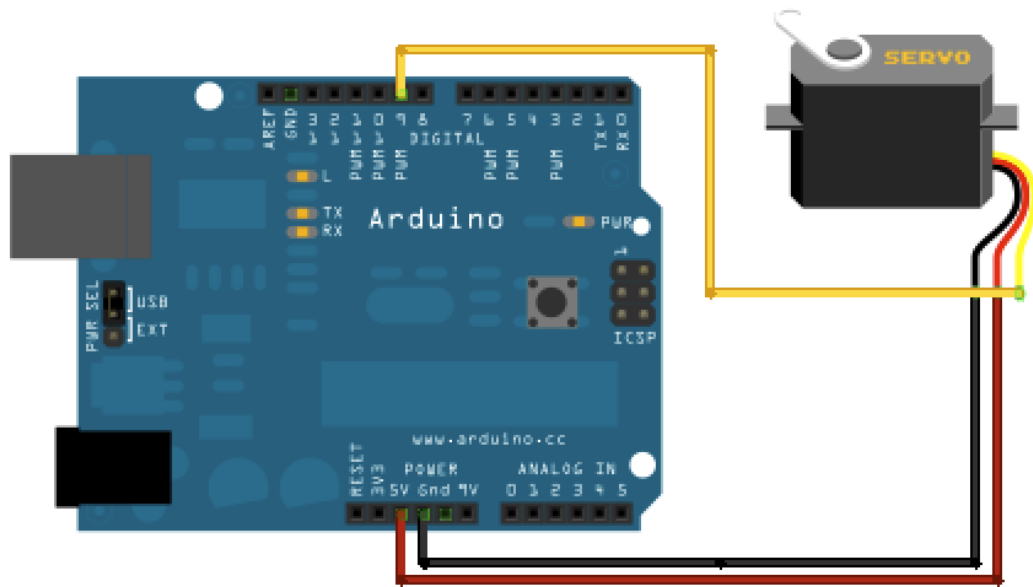
```

right_motor.attach(11); // attaches the servo on pin 11
left_motor.attach(12); // attaches the servo on pin 12
}

void loop()
{
    // Control servo with value 0-180. 90 means that servo stands still
    right_motor.write(0); // 0 - Servo with maximum speed backward
    left_motor.write(180); // 180 - Servo with maximum speed forward
}

```

(vt joonis 13.) (Sell R., 2015)



Joonis 13.

5.6.2 Ülesanne 12. Potentsiomeetriga juhitud servomootor

Kirjeldus: programm demonstreerib potentsiomeetriga juhitud servomootori kasutamist.

```

#include <Servo.h>
Servo right_motor, left_motor; // create servo object to control a servo

void setup()
{
    while (digitalRead(10)==1){}; // Question - What this row does?
    right_motor.attach(11); // Attaches the servo on pin 11
    left_motor.attach(12); // Attaches the servo on pin 12
}

void loop()
{
    int AnalogSensorValue = analogRead(A0); // Read front sensor value
    // Compare it with reference
    if (AnalogSensorValue>500) { // If TRUE

```

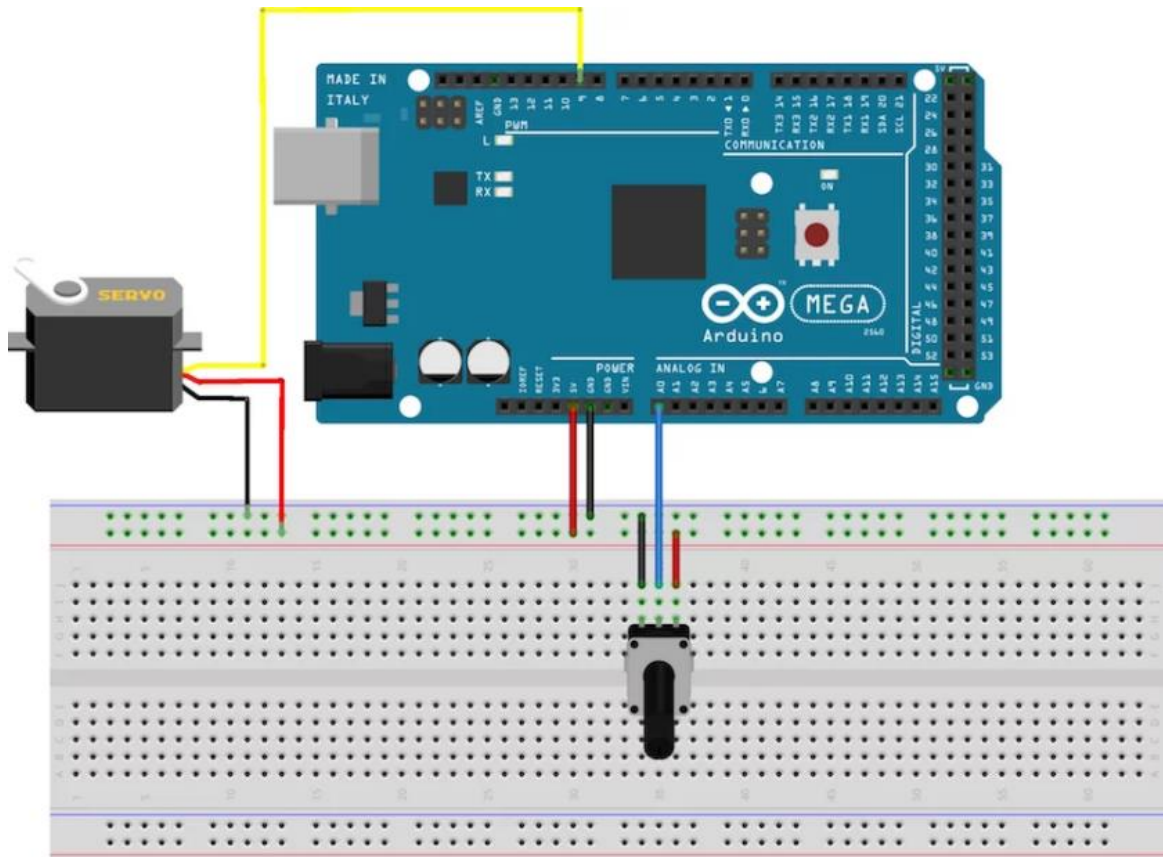


```

    left_motor.write(0); // Reverse one motor
    delay(750); // Wait until turn is enough
  }
  right_motor.write(0); // Drive forward
  left_motor.write(180); // Drive forward
}

```

(vt joonis 14.) (Sell R., 2015)



Joonis 14.

5.7 Kommunikatsioon

5.7.1 Ülesanne 13. Jadaliides

Kirjeldus: programm demonstreerib kommunikatsiooni võimalusi.

```

#include <math.h>
int NTC_sisend = A2; // määrame temperatuurianduri sisendviigu
void setup() {
  Serial.begin(9600);
}

void loop() {
  Termistor(analogRead(NTC_sisend)); // Käivita funktsioon
  Serial.println(""); // Tekita reavahetus
  delay(1000); // Oota 1 sek.
}

```

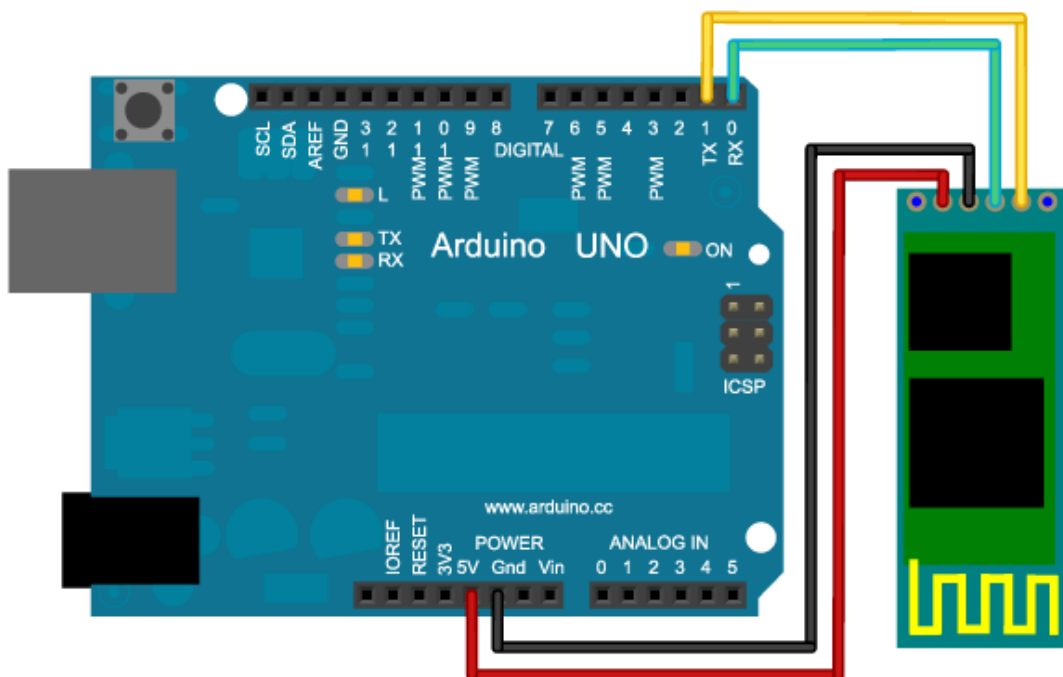
```

double Termistor(int RawADC) {
    double Temp;
    long Takistus;
    // Valem: Takistus = (1024 * JaguriTakisti/ADC) - JaguriTakisti
    Takistus=((10240000/RawADC) - 10000);
    Serial.print("ADC: ");
    Serial.print(RawADC);
    Serial.print("/1024"); // Prindi ADC tulemus
    Serial.print(", Pinge: ");
    Serial.print(((RawADC*5.0)/1024.0),3);
    Serial.print(" volti"); // Prindi ping
    Serial.print(", Takistus: ");
    Serial.print(Takistus);
    Serial.print(" oomi");

    Temp = log(((10240000/RawADC) - 10000));
    Temp = 1/(0.001129148 + (0.000234125 + (0.0000000876741 * Temp *
Temp)) * Temp);
    Temp = Temp - 273.15; // Konverteeri Kelvinid Celciustesse
    Serial.print(", Temperatuur: ");
    Serial.print(Temp);
    Serial.print(" kraadi C");
    return Temp;
}

```

(vt joonis 15.) (Sell R., 2015)



Joonis 15.

6. Lahemduste testimine (õppematerjalide kasutamine praktilises töös)

Ma tegin läbi lõputöös välja toodud ülesanded Tallinna Polutehnikumi õpilastega rühmast IT-15E.

Läbi proovitud juhiste tulemustega sain teada mis tuleks juhistes muuta.

Õpilastel esinesid järgnevad probleemid: LED-id põletati läbi, õpilased võtsid skriptid ja skeemid internetist, mitte ei lahendanud oma peaga.

Lahenduseks seletasin õpilastele kuidas takistid töötavad ja milleks oli neid vaja. Et õpilased ei võtaks skeeme ja skripte internetist, keelasin neil arvutis olles interneti kasutamise.

7. Kokkuvõte

Saime valmis Arduino laboritööd

Tulevikus on lihtsam näiteks õpetajatel õpilastele selgeks teha kuidas Arduino töötab, ning mida on sellega võimalik teha. Sobib ka neile kes soovib ise õppida.

Seda saab edasi arendada nii, et teha veel erinevaid taolisi ülesandeid.

8. Kasutatud kirjandus

Arduino (30.04.2017), Arduino põhiproduktid. [<https://www.arduino.cc/en/Main/Products>
<https://www.arduino.cc/en/tutorial/blink>]

Vikipeedia (30.04.2017), Arduino. [<https://et.wikipedia.org/wiki/Arduino>]

Amatöörrobotika (30.04.2017), Arduino. [<http://robotika.tech-thing.org/2013/moni-sona-arduinost-ja-sellele-koodi-kirjutamisest>]

Roboticlub (01.05.2017), Arduino. <http://home.roboticlub.eu/et/arduino>

Hobilabor (01.05.2017), Arduino. <https://www.slideshare.net/renerebane/arduinooprese-hobilabor2011>

Microsoft (01.05.2017), Arduino IDE. [<https://www.microsoft.com/en-us/store/p/arduino-ide/9nblggh4rsd8>]

9. Lisad