

# Praktilised tööd

---

## Harjutus #1.1

Modifitseerida näiteprogrammi nii, et nupule vajutades vilgub LED (LCD laiendusplaadi korral võib kasutada taustavalguse LED-e) kolm korda.

## Harjutus #1.2

Modifitseerida näiteprogrammi nii, et nupule vajutades hakkab LED konstantselt vilkuma 1 sekundilise intervalliga ja teine nupule vajutus katkestab vilkumise.

## Harjutus #2

Modifitseerida näiteprogrammi nii, et potentsiomeetriga määratakse LED-i heledus, kasutades selleks *delay* funktsioone.

## Harjutus #3.1

Modifitseerida programmi nii, et potentsiomeetriga määratakse temperatuuri nivoo, mis salvestatakse nupuga programmi mällu ja selle nivoo hilisemal ületamisel käivitatakse täiturmehhanism (milleks on LED või ventilaator).

## Harjutus #3.2

Kombineerida LED-i heleduse programm ja eelnev temperatuuri juhtimise programm, kus LED-i heledus/ventilaatori kiirus sõltub sellest, kui palju nivootemperatuuri ületati (1 kraad: heledus 25 %, 2 kraadi: heledus 50 %, 3 kraadi: heledus 75 % ja 5 kraadi: heledus 100 %).

Arduino programmeerimine, koolituse praktilised tööd ja näited, ver. 20151022

Koostas: Raivo Sell

raivo@ittgroup.ee

# Näide #1 Digitaalsed sisendid ja väljundid

---

## Näide 1.1 Nupule vajutades süttib LED

```
/*
Nimetus: Näide #1.1 Digitaalsed sisend-väljundid
Kirjeldus: Programm demonstreerib nupu ja LED-i kasutamist
Autor: Raivo Sell
Kuupäev: 15.01.2013
*/
// Konstandid
const int nupp = A0; // Viik kuhu on ühendatud nupp (teine ots GND)
const int LED = 13; // Viik kuhu on ühendatud 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 == HIGH) { // Kui nupu olek on kõrge (+5 V)
    digitalWrite(LED, LOW); // kustutame LED-i
  }
  else { // vastasel juhul (Maa)
    digitalWrite(LED, HIGH); // süütame LED-i
  }
}
```

## Näide 1.2 Nupule vajutades süttib LED 1 sekundiks

```
// Algas identne näitega #1.1

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
}
```

## Näide 1.3 LED süttib nupu vabastamisel

```
// Algas identne näitega #1.1

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

**NB!** Uuri iseseisvalt nupu sädeluse teemat (*i.k. Button debounce*).

# Näide #2 Analoo sisendid

---

## Näide #2.1 Potentsiomeetri max nivoo ületamisel süttib LED

```
/*
Nimetus: Näide #2.1 Analoo sisend
Kirjeldus: Programm demonstreerib analoogsisendi kasutamist
Autor: Raivo Sell
*/

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

int pote_sisend = A1; // 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
}
```

## Näide #2.2 LED-i vilkumise sagedus sõltub potentsiomeetri sisendist

```
/*
Nimetus: Näide #2.2 Analoo sisendi teisendamine ajaks
Kirjeldus: Programm demonstreerib analoogsisendi kasutamist
Autor: David Cuartielles
Täiendused: Tom Igoe
Kohandused: Raivo Sell
OriginaalURL: http://arduino.cc/en/Tutorial/AnalogInput
*/

int pote_sisend = A1; // 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 <pote> millisekundit
  digitalWrite(led, LOW); // süütame LED-i
  delay(pote); // tekitame viite <pote> millisekundit
}
```

# Näide #3 LCD

---

## Näide #3.1 LCD kasutamine

```
/*
Nimetus: Näide #3.1 LCD kasutamine
Kirjeldus: Kasutab LiquidCrystal.h teeki ja ITEAD Studio 2x16 teksti laiendusplaati
Autor: Tom Igoe
Modifikatsioon: Raivo Sell
Kuupäev: 4.06.2014
Versioon: 1.0
*/
#include <LiquidCrystal.h> // Kaasame vajaliku teegi

// algväärtustame LCD koos vastavate viikude ühendamisega
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

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

}
```

```
/*
Nimetus: Näide #3.2 LCD kasutamine analoogsisendiga
Kirjeldus:
    Kasutab LiquidCrystal.h teeki ja ITEAD Studio 2x16 teksti laiendusplaati
Autor: Tom Igoe
Modifikatsioon: Raivo Sell
Kuupäev: 4.06.2014
Versioon: 1.0
*/
#include <LiquidCrystal.h> // kaasame vajaliku teegi

// algväärtustame LCD koos vastavate viikude ühendamisega
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
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); // tekitame viite 100 millisekundit
}
```

```

/*
Nimetus: Näide #3.3 Temperatuuri andur koos matemaatika teegiga
Kirjeldus:
Kasutab Steinhart-Hart termistori valemit:
temperatuur kelvinites = 1 / {A + B[ln(R)] + C[ln(R)]^3},
kus A = 0.001129148, B = 0.000234125 ja C = 8.76741E-08
Autor: Milan Malesevic and Zoran Stupic
Modifikatsioon: Raivo Sell
Kuupäev: 15.01.2013/4.06.2014
Versioon: 1.1
*/

// kaasame vajalikud teegid
#include <math.h>
#include <LiquidCrystal.h>

// algväärtustame LCD koos vastavate viikude ühendamisega
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
int andur = A5; // määrame anduri 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;

  // 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 pinge väärtuse

  // teine rida
  lcd.setCursor(0, 1);
  lcd.print("R=");
  lcd.print((10240000/RawADC) - 10000); // trükime takistuse väärtuse

  Temp = log(((10240000/RawADC) - 10000));
  Temp = 1/(0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp))* Temp);
  Temp = Temp - 273.15; // konverteerime Kelvin-id Celsius-esse

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

```

$$\frac{1}{T} = A + B \ln(R) + C(\ln(R))^3$$

# Praktilised tööd

---

## Harjutus #4.1

Luaa programm, mis võimaldab servomootori positsiooni või ventilaatori kiirust muuta vastavalt nuppudega seadistatud suurusele.

## Harjutus #4.2

Luaa programm, mis servomootori baasil ehitatud radariga ja fotoanduriga leiaks ruumist kõige eredama punkti.

## Harjutus #5

Luaa programm, mis jadaliidesest saadetud teksti kirjutaks LCD ekraanile. Märk „+“ vahetab rida ning märk „-“ kustutab kogu ekraani.

## Praktiline projekt #1

Koostada lihtne robot, mis suudab navigeerida labürindis ja läbib etteantud tee võimalikult kiiresti.

Kasutatavad vahendid:

- Analoog ja digitaal kaugusandur
- 360 kraadi servomootorid
- Mikrokontroller koos LCD ja nuppude laiendusplaadiga
- Konstruksioonidetailid ja kiirkinnitusvahendid

Arduino programmeerimine, koolituse praktilised tööd ja näited, ver. 20151022

Koostas: Raivo Sell

raivo@ittgroup.ee

## Näide #4 Servomootori juhtimine

```
// Nimetus: Servomootori juhtimine potentsiomeetriga
// Autor: Michal Rinott <http://people.interaction-ivrea.it/m.rinott>
```

```
#include <Servo.h>
```

```
Servo myservo; // Luuakse servo objekt
```

```
int potpin = A1; // potentsiomeetri viigu defineerimine
```

```
int val; // analoogväärtuse lugemi muutuja
```

```
void setup() {
```

```
  myservo.attach(A1); // servo ühendusviik
```

```
}
```

```
void loop() {
```

```
  val = analogRead(potpin); // potentsiomeetri väärtuse lugemine (0-1023)
```

```
  val = map(val, 0, 1023, 0, 179); // väärtuse skaleerimine servomootorile sobivaks vahemikus (0-180)
```

```
  myservo.write(val); // servomootori liigutamine vastavasse positsiooni
```

```
  delay(15); // viide, et servo jõuaks oma liikumise lõpetada
```

```
}
```

## Näide #5 Jadaliides andmete lugemiseks

```
/*
Nimetus: Näide #5 Jadaliidese kaudu saabuvate andmete lugemine
*/
```

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

```
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
```

```
// Globaalsed muutujad
```

```
char data;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600); // Seadistame Serial ühenduse 9600
```

```
  lcd.begin(16, 2); // Määrame LCD read ja veerud
```

```
}
```

```
void loop() {
```

```
  if(Serial.available()) { // Kontrollime, kas on Serial ühendusest tulnud andmeid
```

```
    data = Serial.read(); // Loeme tulnud andmed muutujasse "data"
```

```
    lcd.setCursor(0, 1);
```

```
    lcd.print(data); // Kirjutame saadud sümboli LCDle
```

```
  }
```

```
}
```