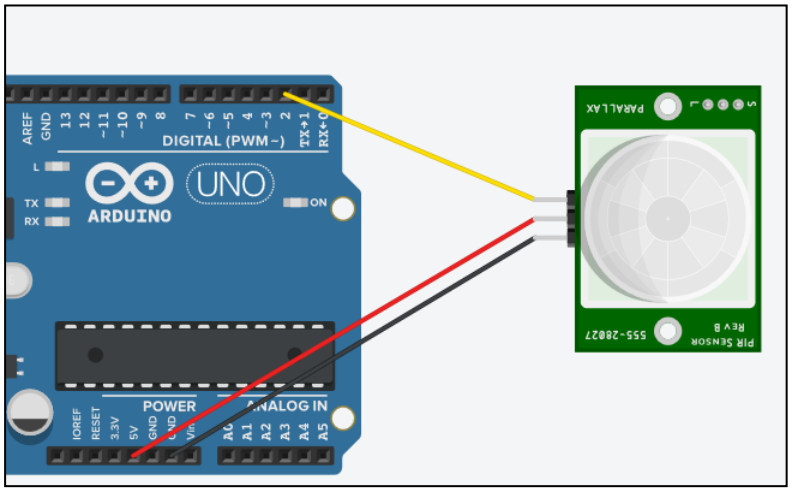
Arduino koolituse kaheksas tund – PIR andur

Õppematerjali koostajad: Indrek Karo ja Angela Leppik

PIR – andur

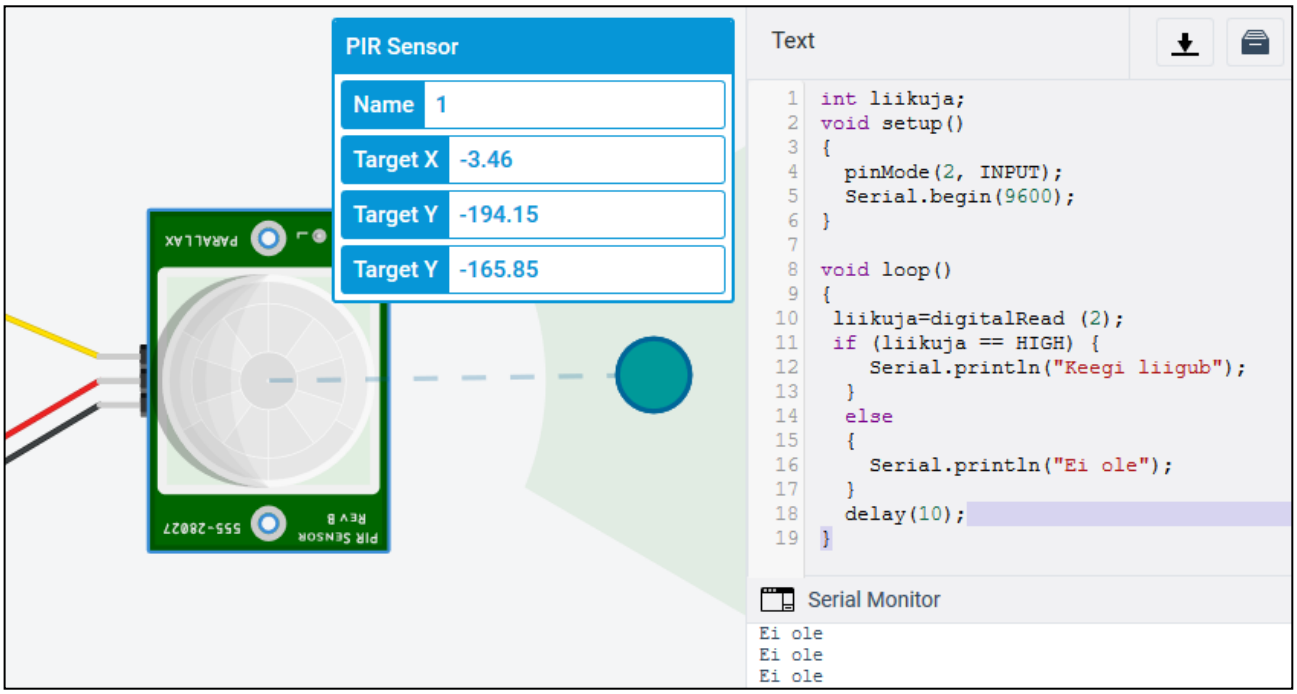
Tegemist on liikumisanduriga, mis tavaliselt töötab kasutades infrapunakiirgust. Kui mingi takistus ilmub ette, siis andur saadab signaali (tavaliselt HIGH). Andur toimib kasutades digitaalsignaali. See on üks tavalisemaid andureid näiteks vargavastastes seadmetes. Pildil nr. 53 on ära toodud üks ühendamise viise Arduinoga.



Pilt nr. 53

Sellel liikumisanduril on kolm jalga: 5 volti, maandus ja signaal (viimane on ühendatud pesasse 2). Tihtipeale kasutatakse sellise anduri puhul anduri tegevuse näitamiseks kas LED-i või siis väljastatakse mingi teade. Pildil nr. 54 on näha liikumisandur töötamas ning nagu jadapordi monitorilt on näha, siis kedagi parajasti anduri lähedal ei ole.

Sinine pallike liikumisanduri ees hallis alas (see on ala, kus liikumisandur on võimeline tegevust tuvastama) on simulaatoris liikuvaks objektiks. Programmis on selle liigutamise näitajaks tehtud muutuja nimega “liikuja”.



Pilt nr. 54

Programmi selgitus

```
int liikuja;                    //luuakse muutuja nimega liikuja
void setup ()
{
 pinMode (2, INPUT);            //oodatakse sisendit pesast 2
 Serial.begin (9600);           //pannakse tööle jadapordi monitor
}

void loop ()
{                               //korduse algus
 liikuja=digitalRead (2);       //muutuja "liikuja" väärtus on pesast 2 loetav digitaalsignaal
 if (liikuja == HIGH) {         //"if" lause, algab tegevus, mis toimub siis kui signaal on
                                HIGH
   Serial.println ("Keegi liigub");  //kui signaal on HIGH, siis jadapordi monitori kirjutatakse
                                "Keegi liigub"
 }                              //„if" lause lõpp
 else                           //kui pesast 2 (muutuja nimega "liikuja" olek ei ole HIGH, siis
```

```
                                toimub järgnev tegevus
 {                              //else algus
  Serial.println ("Ei ole");    //kui liikumist ei tuvastata siis kirjutatakse jadapordi monitori
                                “Ei ole”
 }                              //“else” lõpp
 delay (10);                    //viiteaeg, enne kui järgmist signaali ootama hakkab, et saaks
                                väljastada jadapordimonitoris jälgitavat teksti
}                               //korduse lõpp
```

Teegid ja nende mõte

Teegid on põhimõtteliselt mingi riistvara kasutamiseks teiste inimeste poolt valmis tehtud programmid. Tavaliselt kaasnevad need erinevate seadmetega (servomootorid, ekraanid, mitme mootori kasutamine korraga jne). Teeke saab lisada programmeerimise osas keskmise nupuga (libraries).

Et kasutada LCD ekraani peab kasutama teeki, sest ekraanil on mitmeid ühenduskohti ning nendega kaasnevad erinevad tegevused. Seetõttu ei ole mõistlik ise hakata Arduinole selgeks tegema mida miski ühendus teeb. Tavaliselt on teegi valmis teinud riistvara tootja (sarnaneb loogikalt riistvara draiverile suuremate arvutite puhul). Tavaline teek, mis tuleb kaasa nii Arduino enda tarkvaraga kui ka Tinkercadis, ühildub kõigi LCD ekraanidega, mis kasutavad Hitachi HD44780 draiverit.

LCD ekraani puhul tuleb ühendada kõigepealt vajalikud osad

Praegusel juhul on selleks mõistlik kasutada makettplaati, sest nii on ühendamine lihtsam ja selgem. LCD ekraanil on natuke rohkem väljaviike, kui varem käsitletud seadmetel. Järgnevalt ongi kirjas paremalt poolt alustades erinevate pesade nimed ning nende eesmärk. Ühendust saab vaadata ka pildilt nr. 55.

GND – maandus

VCC – 5 volti ekraani kontrolleri jaoks

V0 – ekraani kontrastsuse kontrollija, kes tahab, saab siia vahele panna näiteks potentsiomeetri või valgusanduri, mis omakorda paneb ekraani heleduse sõltuma välisvalgustuse tugevusest.

RS – siit kontrollitakse kuhu kohta LCD mälus andmed kirjutatakse. Seal saab valida, andmeregistrisse kirjutamise (need andmed, mis lähevad otse ekraanile) või juhendiregistrisse kirjutamise vahel (siit otsib LCD kontroller juhendeid, mida järgmiseks teha). Selle kasutust kontrollitakse teegi abil (teegis on kirjas, kuidas see täpselt toimib) nii, et ekraani enda kasutaja ei pea tegema muud kui ainult asjad ära ühendama.

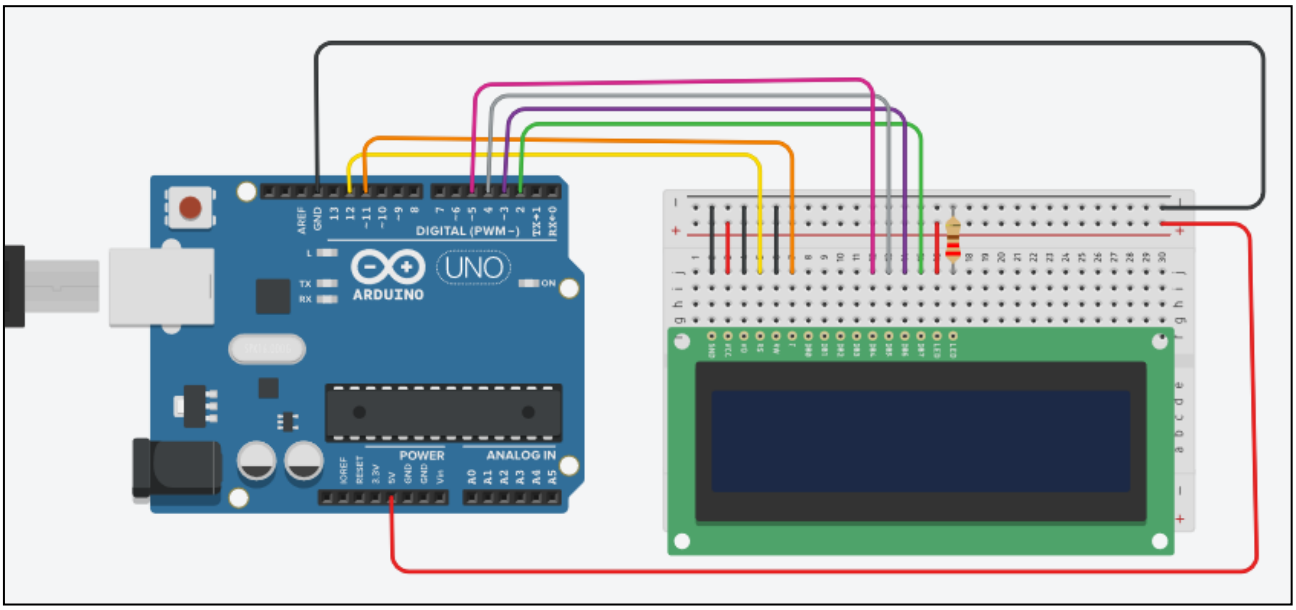
RW – siit valitakse, kas andmeid loetakse või kirjutatakse (sama teema nagu enne, seda kontrollib teek)

E – võimaldab andmeid registritesse kirjutada, kui see ei ole ühendatud, siis andmeid kirjutada ei saa (toimimist kontrollib teek)

DB1-DB7 – nende väljaviikude olekust sõltub see, mida kirjutatakse, kas 1 või 0. Andmed, mida kasutatakse, liiguvad läbi nende pinide.

LED – 5 volti, mida kasutatakse ekraani taustavalgustuseks (muidu pole midagi ekraanil näha)

LED – maandus läbi takisti (et tekiks vooluring), samuti taustavalgustuse jaoks



Pilt nr. 55

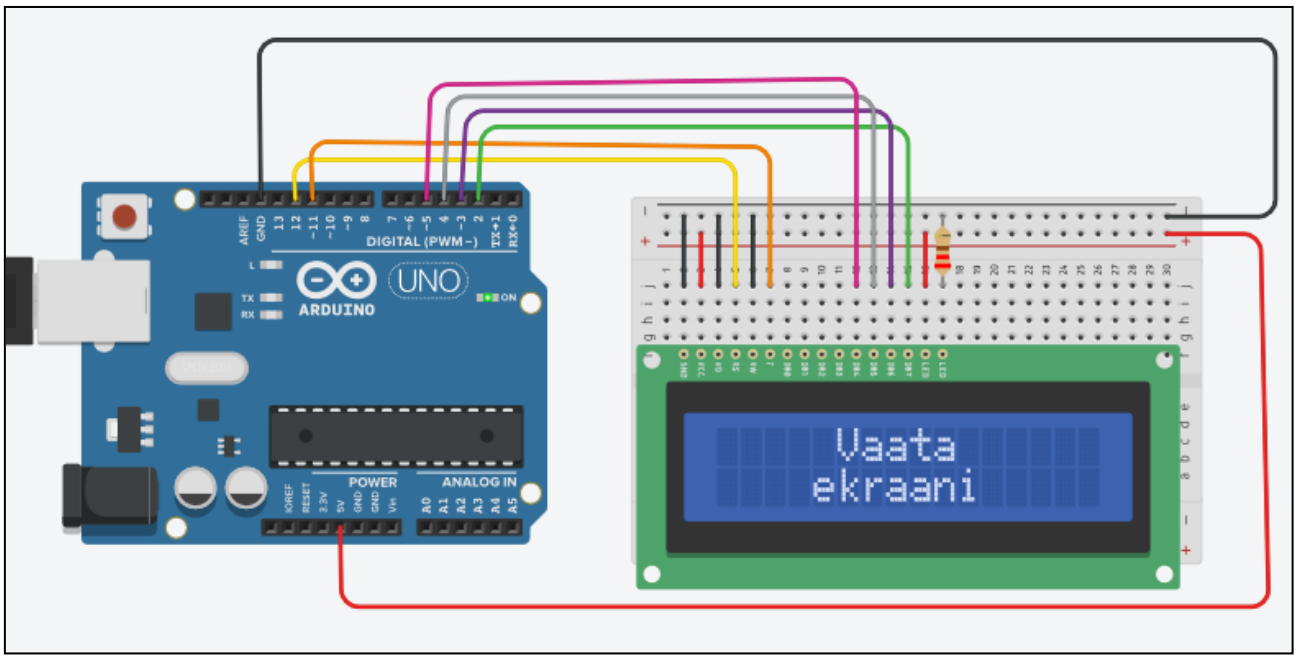
Esimene programm – kirjutame ekraanile natuke juttu

Siin võtame kasutusele kolm uut käsku: lcd.begin (), lcd.setCursor () ja lcd.print (), mis on lahti seletatud programmi kommentaarides. Need käsud on seotud teegiga ehk nad seal ära defineeritud. Ilma teegita ei ole neist käskudest kasu, sest Arduino ise neid käske ei tea.

```
#include <LiquidCrystal.h>                  //lisame teegi
LiquidCrystal lcd (12, 11, 5, 4, 3, 2);     //ütleme milliseid väljaviike kasutab ekraan

void setup ()
{
 lcd.begin (16, 2);                         //alustame ekraani kasutamist ning ütleme, et ekraanil
                                            on 16 veergu ja 2 rida
}

void loop ()
{
  lcd.setCursor (5,0);                      //ütleme, et kirjutamist alustatakse 5 veerust (seda
                                            tähistab "5" ja esimesest reast (seda tähistab "0" ehk
                                            rida nr. 0, arvutid alustavad loendamist alati 0-st)
  lcd.print ("Vaata");                      //kirjutatakse string "Vaata"
  lcd.setCursor (4,1);                      //ütleme, et nüüd on vaja kirjutama hakata 4 veerust ja
                                            teisest reast (seda tähistab "1" ehk rida nr. 1)
  lcd.print ("ekraani");                    //kirjutatakse string "ekraani"
}
```



Pilt nr. 56

Nüüd näitab Arduino ekraanil kahel real teksti. Vaata ekraani pildil nr. 56

Teksti vahetamise programm

Kuna teksti ekraanil saab muuta, siis on võimalik kirjutada programm, mis vahetab teksti vastavalt sellele, mida on vaja näidata. Siin saaksime teha programmi, mis vahetab teksti alumist osa.

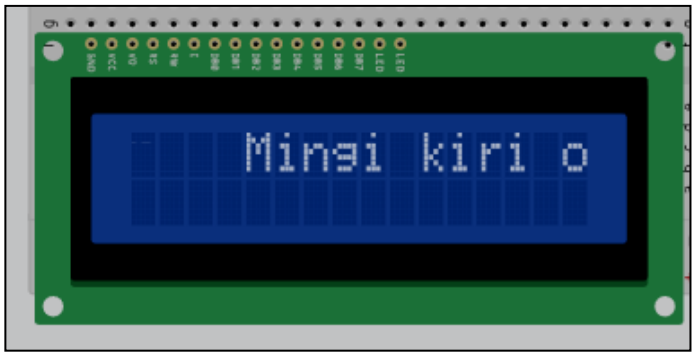
```
#include <LiquidCrystal.h>                  //lisame teegi
LiquidCrystal lcd (12, 11, 5, 4, 3, 2);     //ütleme milliseid väljaviike kasutab ekraan

void setup ()
{
 lcd.begin (16, 2);                         //hakkame kasutama ekraani 16. veergu ja 2. rida
 lcd.print ("Tere");                        //kirjutame kohe alguses “Tere”, siis ei pea pärast
                                            muutma
}

void loop ()
{
  lcd.setCursor (0,1);                      //hakkame kirjutama teise rea algusesse
  lcd.print ("ekraani pealt");              //kirjutame “ekraani pealt”
  delay (1000);                             //ootame ühe sekundi
  lcd.setCursor (0,1);                      //hakkame uuesti kirjutama teise rea algusest
  lcd.print ("          ");                 //kirjutame tühjust (vähemalt samapalju kui eelnevat
                                            teksti), kui tühjust ei kirjuta, siis üritab Arduino
                                            kirjutada kahte teksti samasse kohta
  delay (100);                              //ootame 100 millisekundit
  lcd.setCursor (0,1);                      //hakkame uuesti kirjutama teise rea algusest
  lcd.print ("siitpoolt");                  //kirjutame teksti “siitpoolt”
  delay (1000);                             //ootame ühe sekundi, et teksti lugeda jõuaks
  lcd.setCursor (0,1);                      //hakkame uuesti kirjutama teise rea algusest
  lcd.print ("          ");                 //kirjutame jälle tühjust, et saaks uut teksti näidata
  delay (100);                              //ootame 100 millisekundit, et tekst jõuaks ära kaduda
}
```

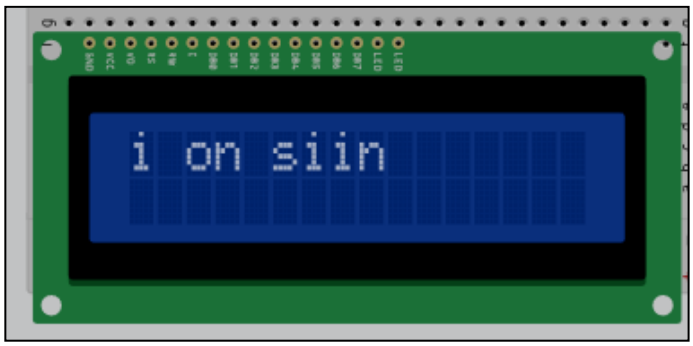
Ekraanil liikuv tekst

Ekraanil saab teksti panna ka liikuma mingis suunas. Sellise võimaluse kasutamine on hea siis, kui tekst on liiga pikk ning ei mahu korraga ekraanile ära. Ekraanil teksti liigutamise puhul on vaja kasutada paari uut käsku nimega lcd.scrollDisplayLeft (); ja lcd.scrollDisplayRight ();, mis panevad teksti liikuma vastavalt vasakule või paremale. Selline tegevus on näha ka piltidel nr. 57, 58 ja 59.



Pilt nr. 57

Pilt nr. 58



Pilt nr. 59

Teksti liigutamise programm

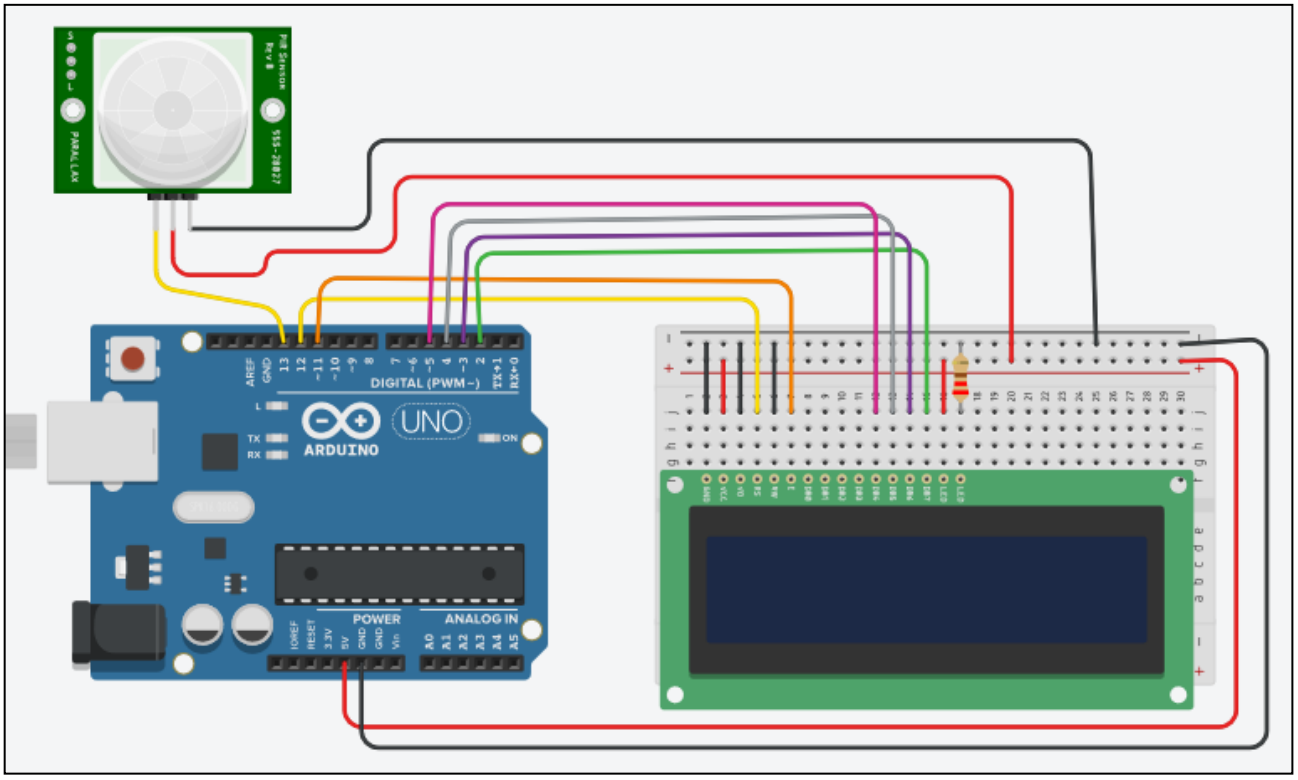
```
#include <LiquidCrystal.h>          //tutvustame Arduinole LCD ekraani teeki
LiquidCrystal lcd (12, 11, 5, 4, 3, 2); //ütleme Arduinole, milliseid väljaviike kasutab LCD

void setup ()
{
lcd.begin (16, 2);                  //ütleme Arduinole mitu veergu ja rida on ekraanil
  lcd.print ("Mingi kiri on siin");  //ütleme Arduinole millist teksti näidata
  delay (1000);                     //ootame 1 sekundi, sest muidu on teksti raske jälgida
}

void loop ()
{
  for (int asukoht = 0; asukoht < 19; asukoht++) {   //tekitame “for” tsükli ning kasutame muutujat
                                                     “asukoht”, mis muutub 0 kuni 19, liigutamaks
                                                      teksti ühe ühiku võrra
  lcd.scrollDisplayLeft ();         //keri üks ühik vasakule
  delay (150);                      //ootab natuke, muidu on halb jälgida
  }
}
```

Loendur ekraanil – programm, koos PIR anduriga

Ekraani saab kasutada ka mitme erineva anduriga koos. Pildil nr. 60 on ära toodud PIR anduri kasutamine koos ekraaniga, näitamaks kas keegi on anduri läheduses liikumas. Sama skeemi saab kasutada ka muude programmide jaoks, näiteks loenduri tegemiseks.



Pilt nr. 60

Antud süsteem on sarnane varem vaadatutega ning seetõttu on ka programmi kombinatsioon varem kasutatud programmidest.

```
#include <LiquidCrystal.h>          //tutvustame Arduinole LCD ekraani teeki
LiquidCrystal lcd (12, 11, 5, 4, 3, 2); //ütleme Arduinole, milliseid väljaviike kasutab LCD

void setup ()
{
 pinMode (13, INPUT);               //pesa 13 on liikumisanduri sisendiks
 lcd.begin (16, 2);                 //ütleme Arduinole mitu veergu ja rida on ekraanil
}

void loop ()
{                                   //korduse algus
  if (digitalRead (13) == 1){       //kui liikumisanduri signaal on “1”, siis toimub järgnev
    lcd.setCursor (0, 1);           //hakkame kirjutama teise rea algusesse
    lcd.print ("keegi on siin");    //kirjutab “keegi on siin”
```

```
 }                                  //"if" tsükli lõpp
 else{                              //muud juhud (kui liikumisandur on "0")
   lcd.setCursor (0,1);             //hakkame kirjutama teise rea algusesse
         lcd.print ("ei ole kedagi"); //kirjutab "keegi on siin"
   }                                //"else" lõpp
}                                   //korduse lõpp
```

Loendur ekraanil

Ekraanil loenduri näitamiseks on arukas kasutada paari muutujat, mille abil saab näidata, millises vahemikus numbrid muutuvad. Antud programmis on kasutusel muutujad "loendur", mille väärtus muutub (hetkel väheneb alates 10-st) ning muutuja "kuni", mis näitab kui kaugele loendur loendab (hetkel 0).

```
#include <LiquidCrystal.h>          //tutvustame Arduinole LCD ekraani teeki
LiquidCrystal lcd (12, 11, 5, 4, 3, 2); //ütleme Arduinole, milliseid väljaviike kasutab LCD

void setup ()
{
 lcd.begin (16, 2);                 //ütleme Arduinole mitu veergu ja rida on ekraanil
 lcd.print ("Hakkan karjuma");      //esialgne teave, mis ilmub kohe ekraanile
}

void loop ()
{
 int loendur = 10;                  //loenduse algus, annab muutuja "loendur" väärtuseks "10"
  int kuni = 0;                     //loenduse lõpp, annab muutuja "kuni" väärtuseks "0"
  while (loendur>=kuni){            //tekitatakse "while" tsükkel, mis toimib senikaua kuni
                                    muutuja "loendur" on suurem või võrdne muutujaga "kuni"
   lcd.setCursor (0, 1);            //alustab teiselt realt ekraani algusest
   lcd.print ("          ");        //kirjutab tühjust
   delay (10);                      //ootab, sest muidu toimuks tegevus liiga kiiresti
   lcd.setCursor (0,1);             //läheb teisele reale ekraani algusesse
   lcd.print (loendur);             //näitab numbrit (muutuja "loendur" hetkelist väärtust")
```

```
  loendur = loendur-1;          //vähendab loenduri väärtust ühe võrra
  delay (500);                  //ootab pool sekundit
 }                              //“while” tsükli lõpp
 lcd.setCursor (0, 1);          //alustab teiselt realt ekraani algusest
 lcd.print ("Hirmus kisa!!");   //kirjutab “Hirmus kisa!!”
 delay (3000);                  //ootab 3 sekundit, et alustada uuesti loenduriga algusest
}                               //korduse lõpp
```