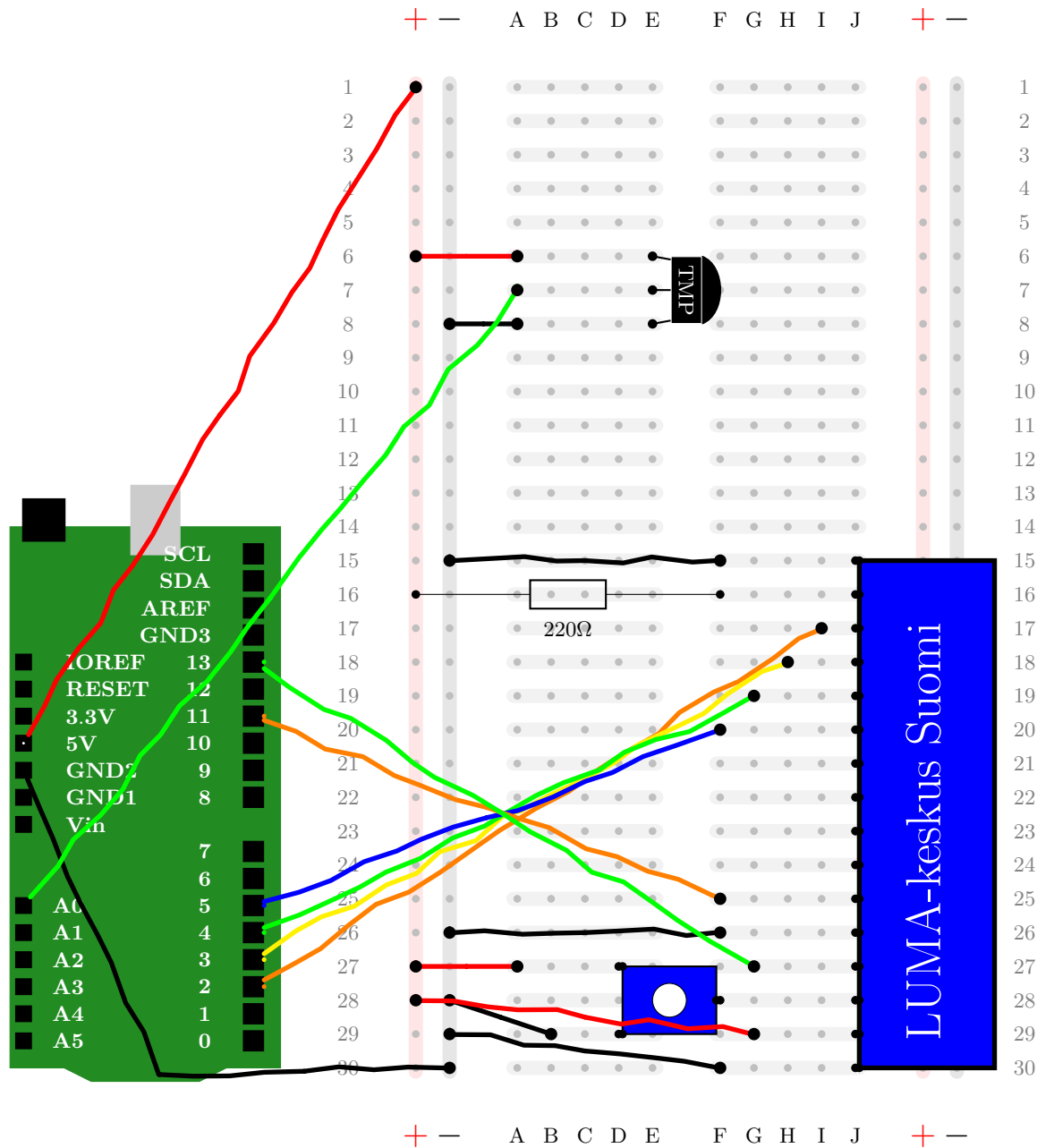


Arduino-paketti yläkouluun ja lukioon



Käyttölisenssi

Tämä materiaali rakentaa ja muokkaa Arduinon Projects Book:n ohjeita, jotka on lisensoitu Nimeä-EiKaupallinen-JaaSamoin 3.0 Ei sovitettu (CC BY-NC-SA 3.0) (Attribution-NonCommercial-ShareAlike 3.0 Unported (CC BY-NC-SA 3.0)).

Alkuperäinen tekijä: arduino.cc

Muokkaajat: Heli Virtanen ja Pekka Muotka

Materiaalin testaajat ja kommentoijat: Janne Rantanen ja Ilkka Palonen

Alkuperäisen materiaalin lisenssin mukaisesti, myös tämän materiaalipaketin materiaaleja voi käyttää lisenssin CC BY-NC-SA 3.0 mukaisesti.

This work is licensed under a [Creative Commons "Attribution-NonCommercial-ShareAlike 3.0 Unported"](https://creativecommons.org/licenses/by-nc-sa/3.0/) license.



Lisäksi ideoita ja inspiraatiota materiaalin pohjaksi ovat antaneet LUMA-keskus Saimaan sekä Aalto-yliopiston Arduino-materiaalit.

<https://act.aalto.fi/fi/aalto-yliopisto-junior/materiaalit>

Kiitokset myös tiedekasvatuskurssin opiskelijoille L. Sihvonen ja M. Moilanen.

Taustaa

Tämä materiaali on tehty LUMA-keskus Suomen sekä kehittäjäkoulujen kanssa yhteistyössä lukuvuoden 2021-2022 aikana.

Materiaalia kehitettiin koulujen toiveiden pohjalta ja kehittämissyhteistyö jatkuu edelleen.

Paketista julkaistaan kesän 2022 LUMA-päivien yhteydessä versio, jota tullaan myöhemmin täydentämään. LUMA-keskus Suomen sivuilta löytyy koko materiaali käyttövalmiina PDF-tiedostona ja lisäksi siellä on linkki kehitysversioon. Tämä versio on tarkoitettu erityisesti sähköisesti käytettäväksi ja täytettäväksi, mutta toimii myös paperiversiona.



**LUMA-KESKUS
SUOMI**

Sisällys

1 Koekytkentälevyn käyttö	4
1.1 Rinnankytkentä	4
1.2 Sarjaankytkentä	5
1.3 Useampi kaksijalkainen komponentti piirissä	6
1.4 Koekytkentälevystä	10
1.5 Ensimmäinen kytkentä koekytkentälevylle	11
1.6 Lisätään kytkentään painonappi	15
2 Liikennevalot	17
2.1 Ohjelmoitava valo	17
2.2 Ohjelmoitava liikennevalo	21
2.3 Painonapin lisääminen	25
2.4 Suunnittelu omat liikennevalot risteykseen	34
3 Lämpötila-anturi ja valoanturi	36
3.1 Lämpömittari	36
3.1.1 Lämpötila-anturin käyttöönotto	36
3.1.2 LEDien lisääminen	39
3.2 Valoanturi	44
3.2.1 Valoanturin käyttöönotto	44
3.2.2 LEDien lisääminen	46
3.3 Valo- ja lämpöanturit	49

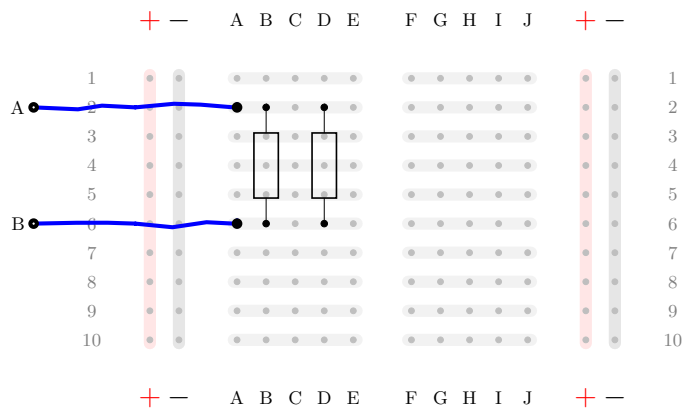
4 LCD-näyttö	53
4.1 LCD-näytön kytkentä	53
4.2 Lämpötilan mittaaminen ja näyttäminen näytöllä	62
4.3 LCD-näyttö ja vaihtuvat viestit	64
A Komponenttipankki	70
B Vinkit Arduinolla koodaamiseen	75
C Vastusten värikoodit	79

1. Koekytkentälevyn käyttö

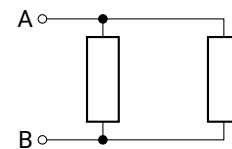
1.1 Rinnankytkentä

Kahden kaksijalkaisen komponentin rinnankytkentä saadaan aikaan laittamalla molempien komponenttien toinen jalka yhdelle riville ja toiset jalat jollekin muulle riville:

Koekytkentälevyllä

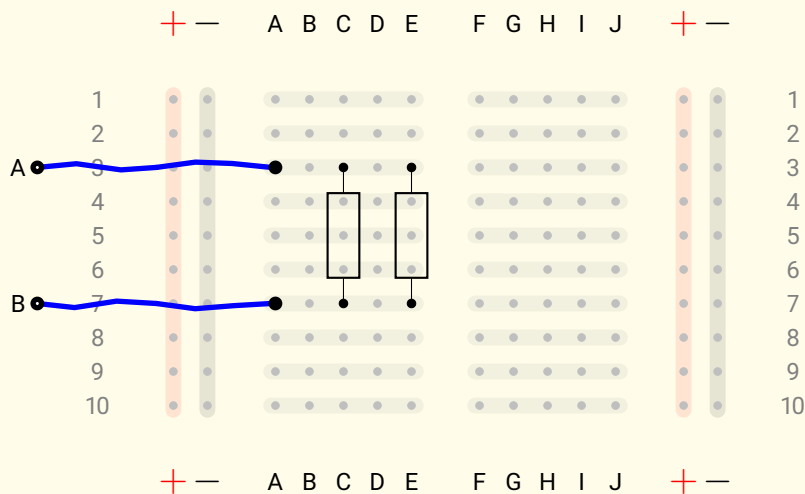


Piirikaavio



Testaa taitosi!

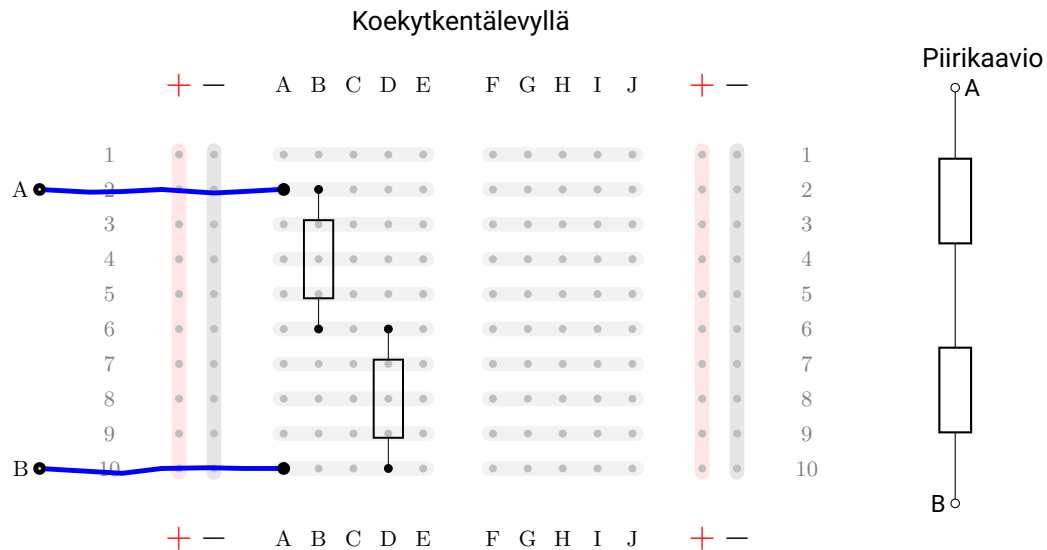
Piirrä toinen kaksijalkainen komponentti jo piirretyn komponentin rinnalle. Piirrä myös pisteet A ja B kuvaan.



Kuvassa on esitetty yksi esimerkki, mutta myös muita vaihtoehtoja on olemassa.

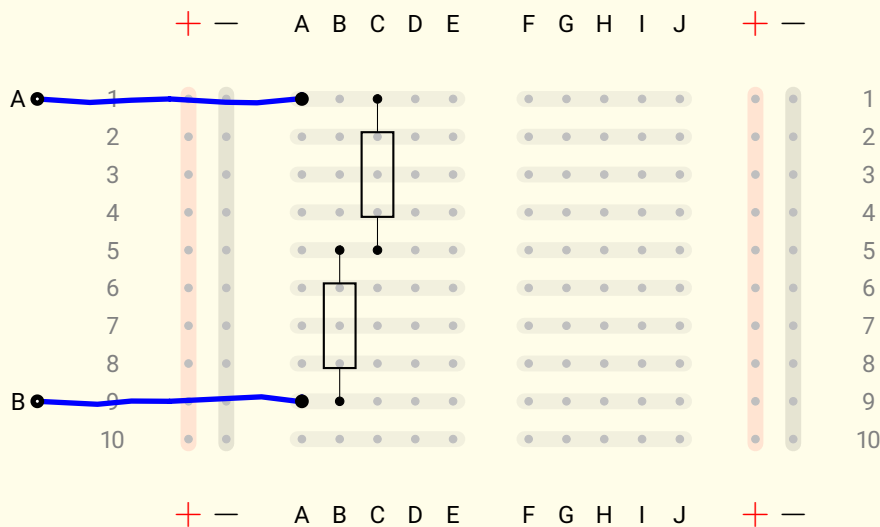
1.2 Sarjaankytkentä

Kahden kaksijalkaisen komponentin sarjankytkentä saadaan aikaan laittamalla molempien komponenttien toinen jalka yhdelle riville ja toiset jalat kahdelle muulle riville:



Testaa taitosi!

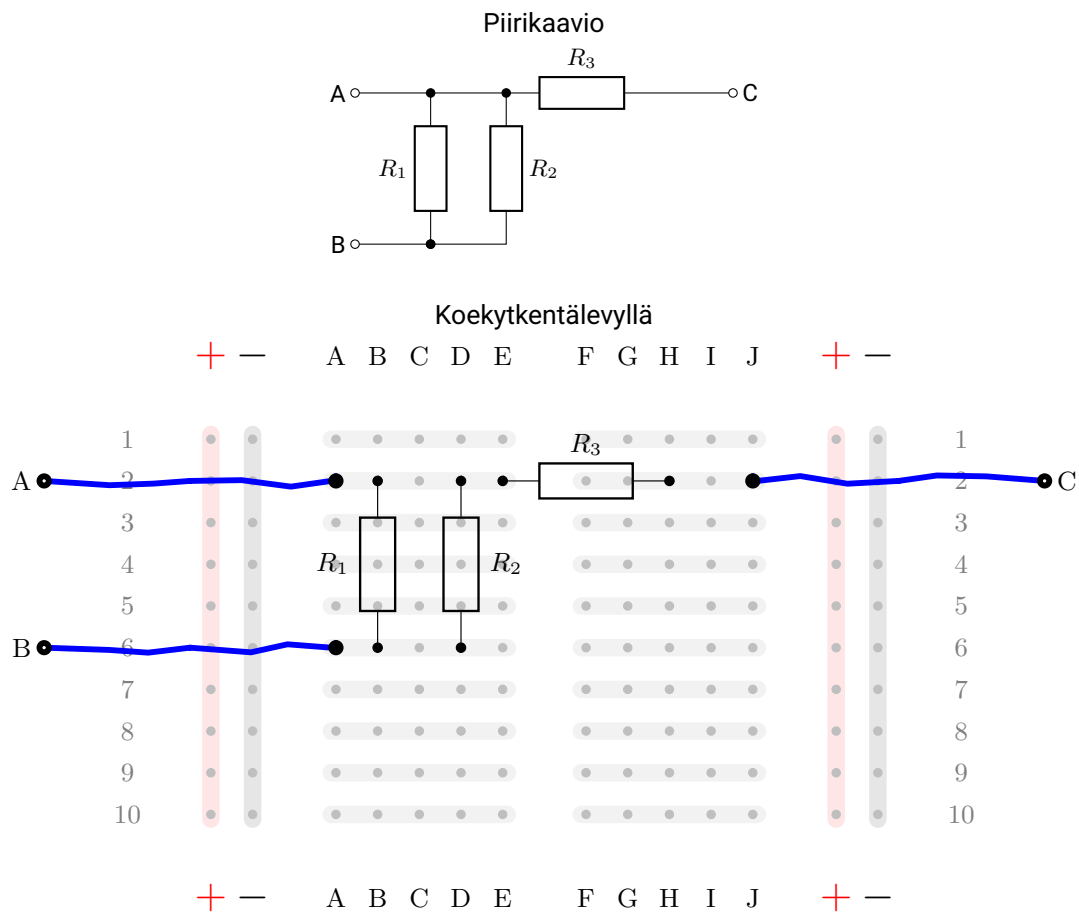
Piirrä toinen kaksijalkainen komponentti jo piirretyn komponentin kanssa sarjaan. Piirrä myös pisteet A ja B kuvaan.



Kuvassa on esitetty yksi esimerkki, mutta myös muita vaihtoehtoja on olemassa.

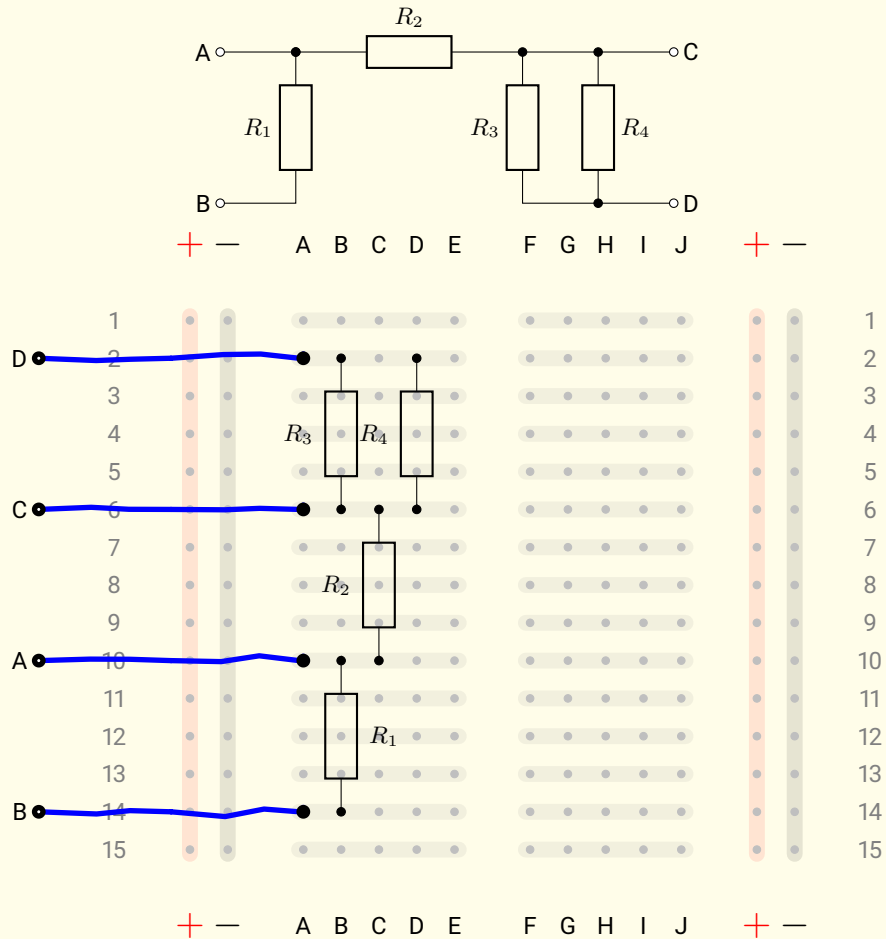
1.3 Useampi kaksijalkainen komponentti piirissä

Harjoitellaan vielä useamman komponentin kytkentää koekytkentälevylle. Nyt, jotta pysytään paremmin perillä, käytetään numeroita komponenttien vieressä, jotta muistetaan mikä komponentti on kyseessä.



Testaa taitosi!

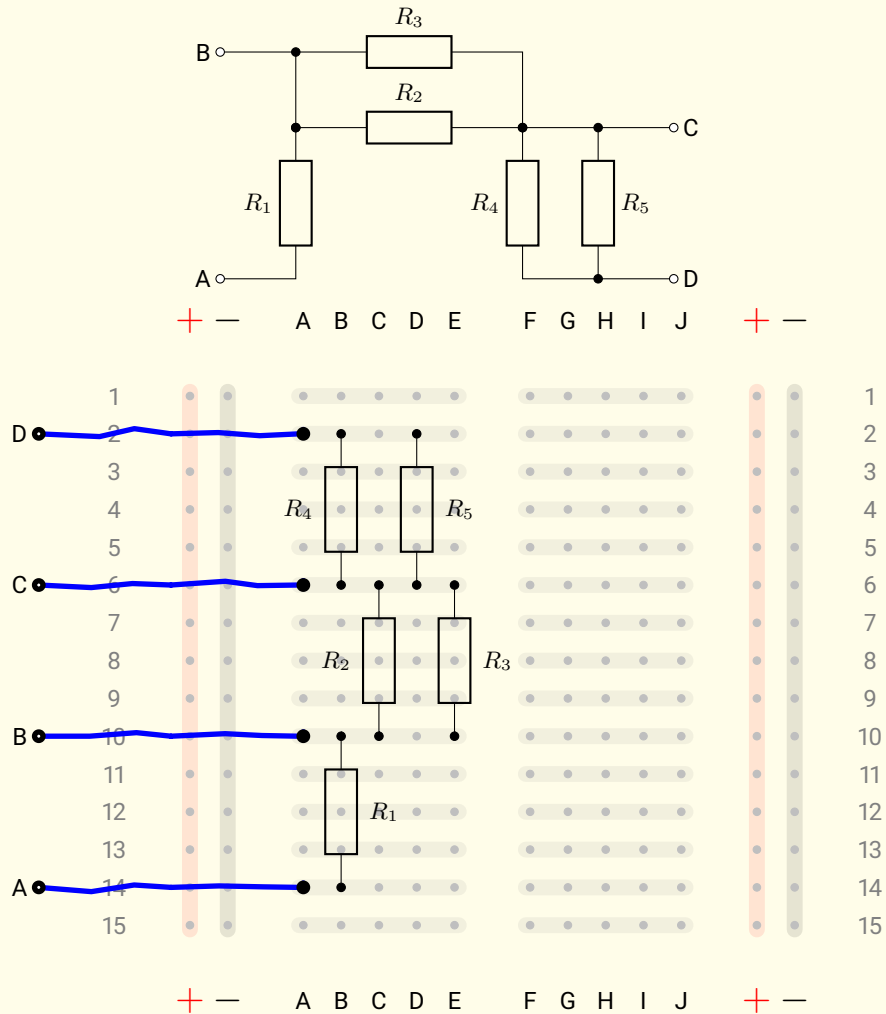
Piirrä kuvan mukainen kytkentä koekytkentälevylle.



Kuvassa on esitetty yksi esimerkki, mutta myös muita vaihtoehtoja on olemassa.

Testaa taitosi!

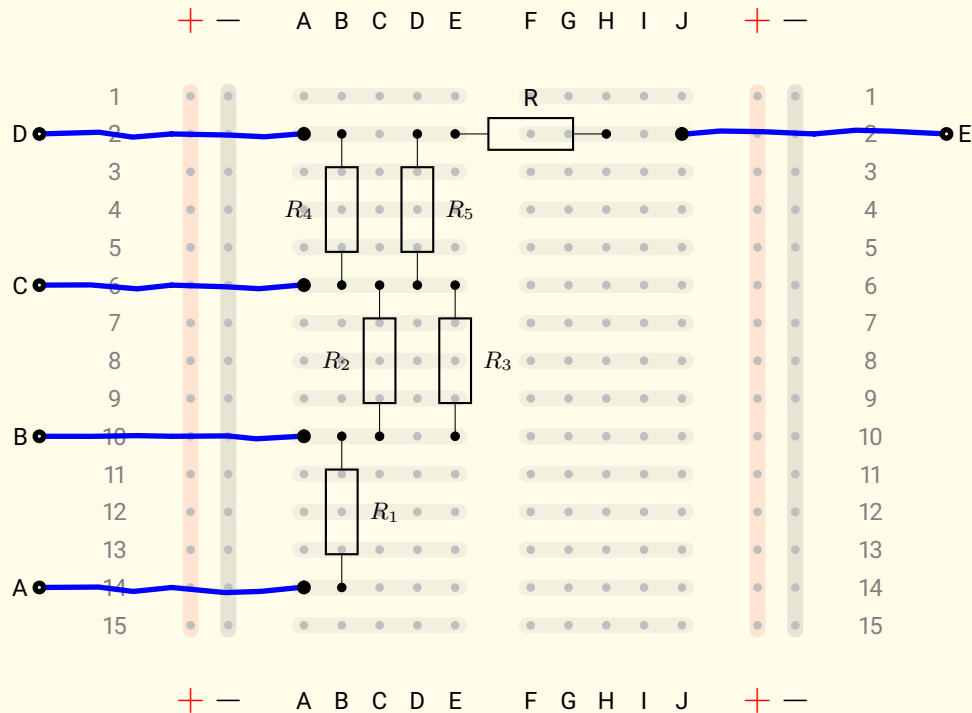
Merkitse koekytkentälevylle johdot solmuihin A, B, C ja D.



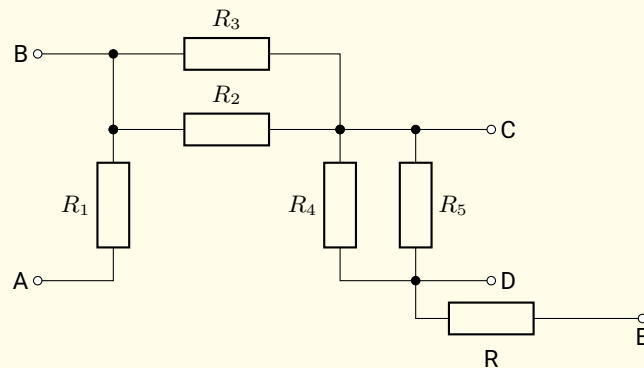
Kuvassa on esitetty yksi esimerkki, mutta myös muita vaihtoehtoja on olemassa.

Testaa taitosi!

Jos edelliseen piiriin pitäisi lisätä vastus olemassa olevan solmun D ja uuden solmun E välille, mihin väliin kytkisit uuden vastuksen?

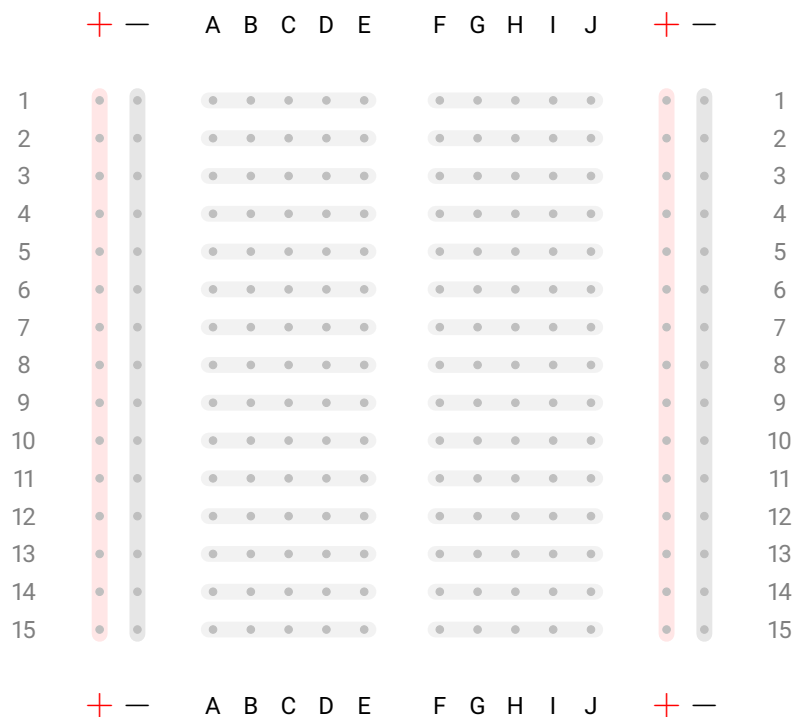


Kuvassa on esitetty yksi esimerkki, mutta myös muita vaihtoehtoja on olemassa.



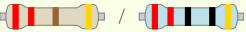
1.4 Koekytkentälevystä

- Yksi rivi (A-B-C-D-E) tai (F-G-H-I-J) vastaa yhtä solmua piirroksessa.
- Jos yhteen solmuun kiinnittyy enintään viisi (5) viivaa, riittää käyttää kaikki valmiina olevat paikat (A-E tai F-J).
- Jos yhdessä solmussa on yli viisi (5) viivaa, tee lisä tilaa lisäämällä johto (eli oikosulku) esimerkiksi saman rivin toiselle puolelle (esimerkiksi käytössä rivi 3: kytke johto E3:sta F3:een, ja sinulla on nyt vapaana A3, B3, C3, D3, G3, H3, I3 ja J3).
- + tai – pystyrivejä käytetään yleensä käyttöjännitteelle (punainen +) sekä maalle (musta –). Näistä lisää seuraavaksi.



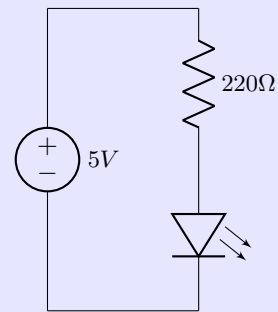
1.5 Ensimmäinen kytkentä koekytkentälevylle

Tarvikkeet

- Yleismittari (vastuksen arvon selvittämiseen)
- Vastus 220Ω : 
- Koekytkentälevy
- Arduino UNO (jännitelähteenä)
- LED
- Hyppylankoja

Piirin toiminta

LEDin harjoituskytkentä. LED palaa, kun piiri on kytketty jännitelähteeseen.



HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

LEDin kanssa kytketään aina sarjaan vastus (nyt 220Ω), kun kytket LEDin $5V$ -jännitteeseen.

Vastuksen arvo

Jos sinulla ei ole käytössä yleismittaria oikean vastuksen arvon mittaamiseen, tai haluat harjoitella vastusten värikoodien lukemista, vastusten värikoodit löydät sivulta 79. Nyt tarvitaan 220Ω vastus eli jompikumpi alla olevista vaihtoehdoista:

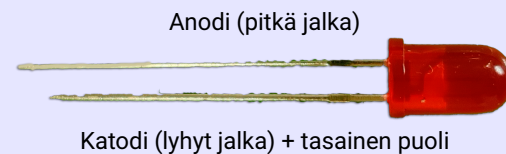
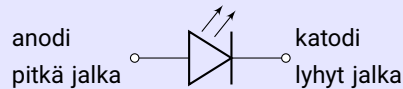


LEDin kytkeminen

Huomaa, että on merkitystä kummin päin LED kytetään koekytkentälevylle!

LEDissä on kaksi jalkaa, joista toinen on pidempi. LEDin muovi on toiselta puolelta kaareva, ja toiselta siinä on suora leikkaus, lyhyempi jalka on suoran leikkauksen puolella. Lyhyemmän jalan nimi on katodi ja pidemmän jalan anodi.

Piirrosmerkissä:

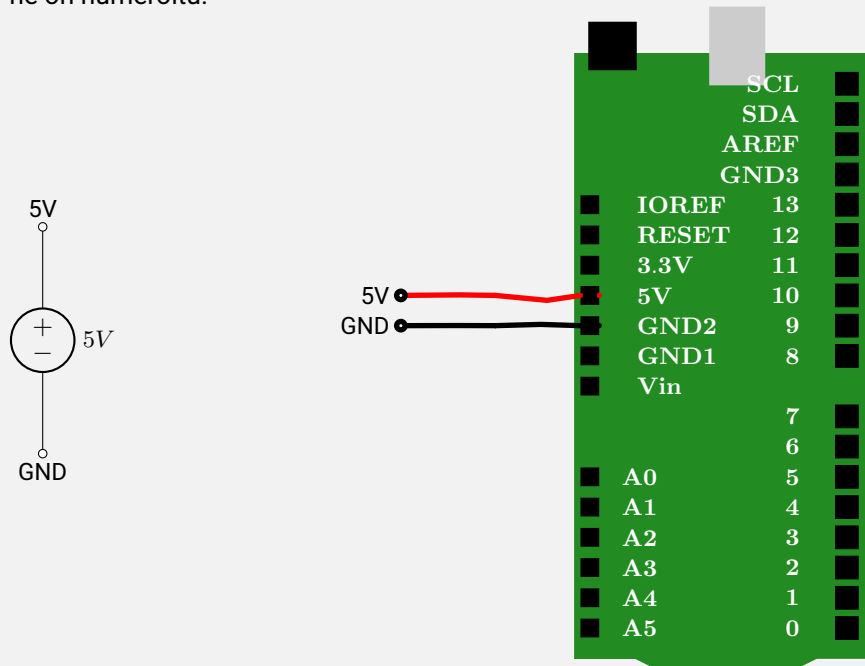


LED, eli valoa säteilevä diodi päästää virtaa läpi vain toiseen suuntaan kuten diodit. Jos anodille kytketään tarpeeksi suuri jännite verrattuna katodiin, virta kulkee ja valo palaa. Jos taas katodilla on suurempi jännite kuin anodilla, virtaa ei kulje ja LED ei pala. Jos anodilla ei ole tarpeeksi suuri jännite verrattuna katodiin, virta ei kulje. Riippuen LEDistä, vaadittu kynnysjännite voi vaihdella ja on tyypillisesti 1,5-4,5 voltia.

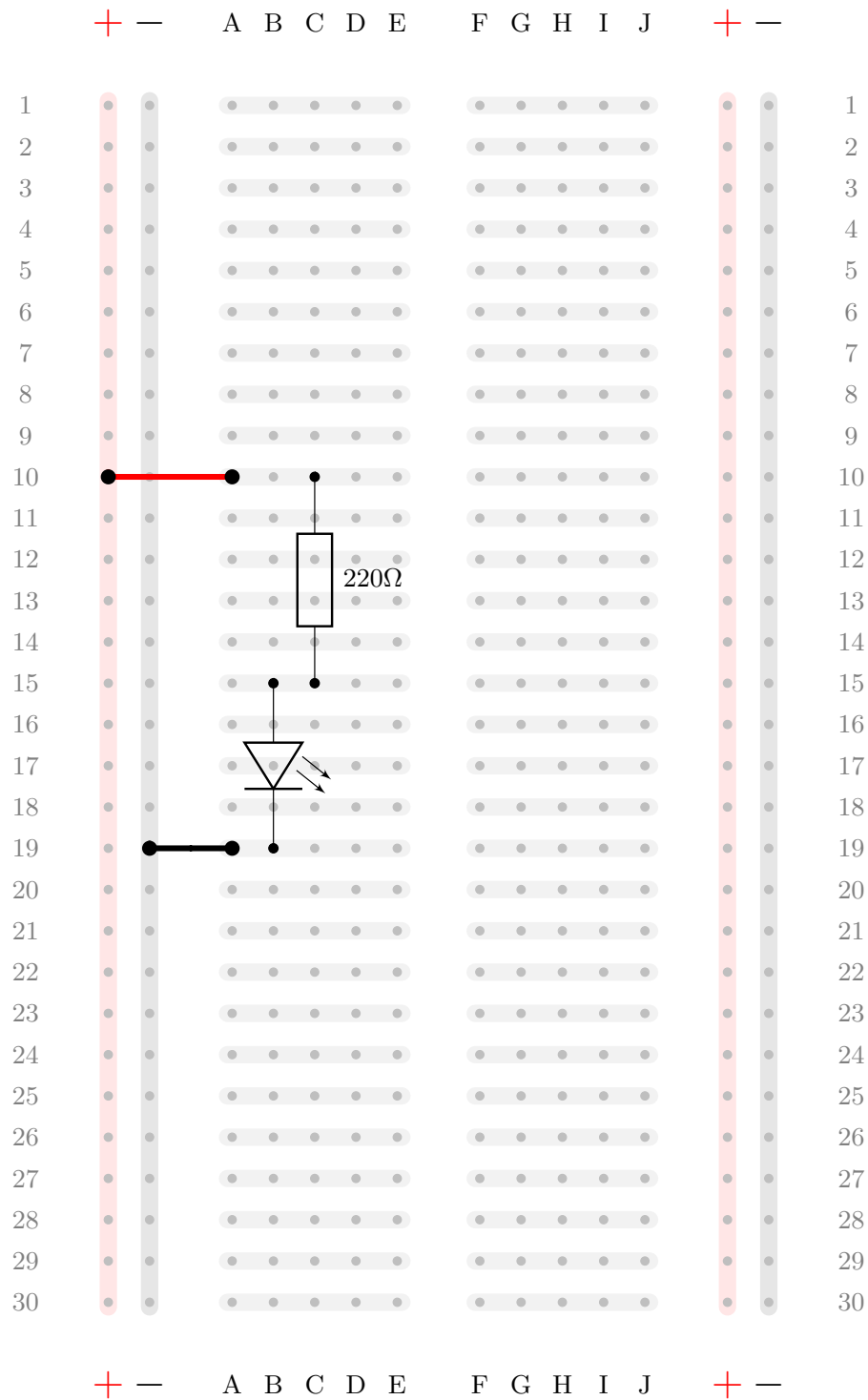
Jännitelähteen kytkeminen

Jännitelähde on Arduinossa sisäänrakennettuna. Jotta voimme kytkeä jännitelähteen piiriin, tarvitsemme jännitelähteen molemmat päät: +5V ja maan (GND), kytkemällä vain toinen piiriin emme saa aikaan haluttua vaikutusta.

Arduino levyllä on kolme eri GND-riviä, ja mikä tahansa niistä käy. Kuvassa selvyiden vuoksi ne on numeroitu.

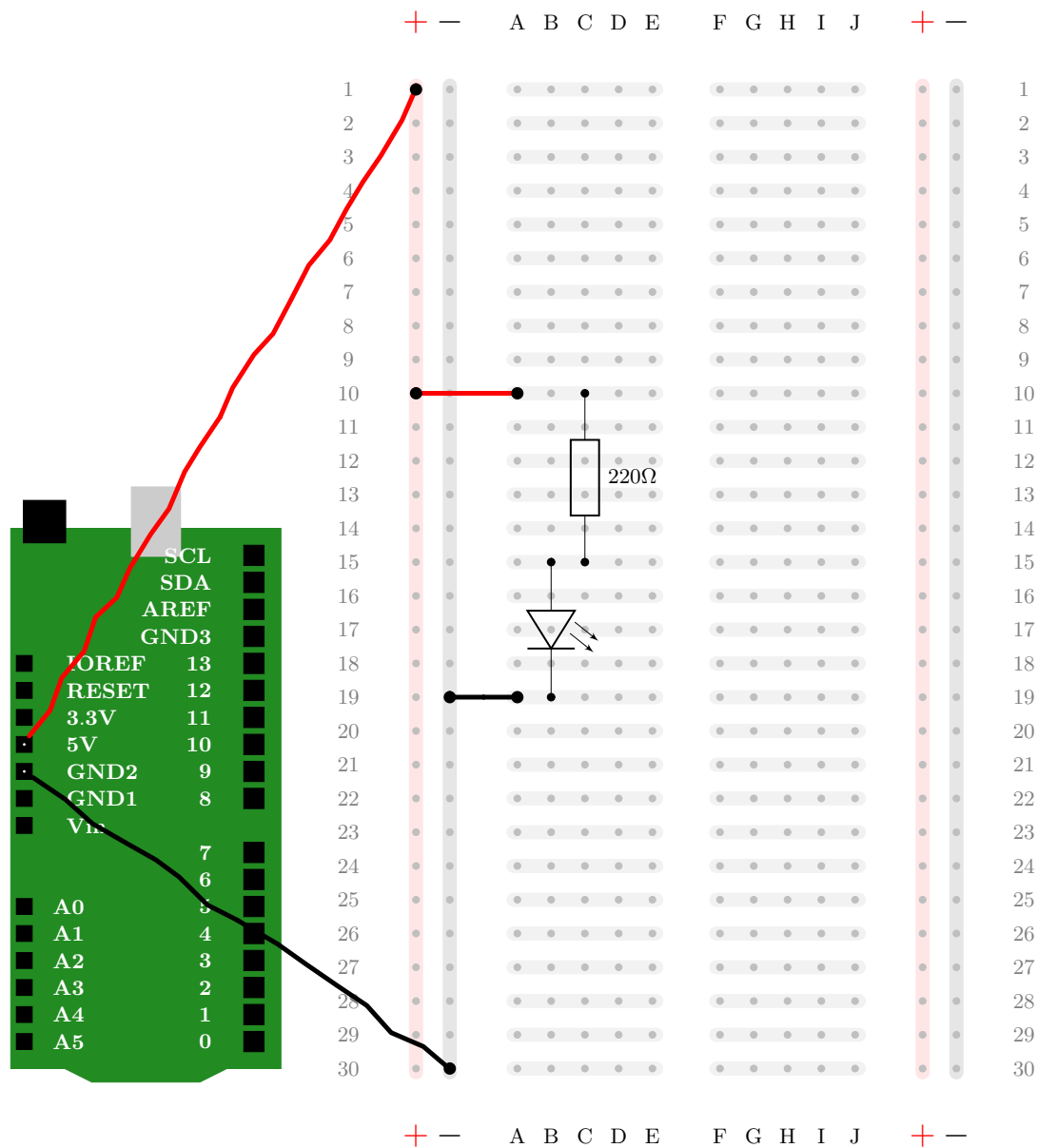


Kytetään ensin vastus ja LED kuvan mukaisesti ja lisätään kaksi hyppylankaa.



Nyt on siis tehty muuten kytkennät ja pitää kiinnittää vielä jännitelähde piiriin.

Kytetään nyt käyttöjännite (+5V) vasemmalle + pystyriville ja maa (GND) tämän viereiselle – pystyriville.



Nyt kun yhdistetään Arduino johdolla tietokoneeseen, LED syttyy!

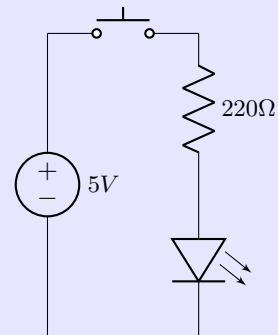
1.6 Lisätään kytkentään painonappi

Tarvikkeet

- Edellisen työn koottu piiri
- Painonappi

Piirin toiminta

Painonapin koekytkentä. Nappia painamalla LED syttyy.



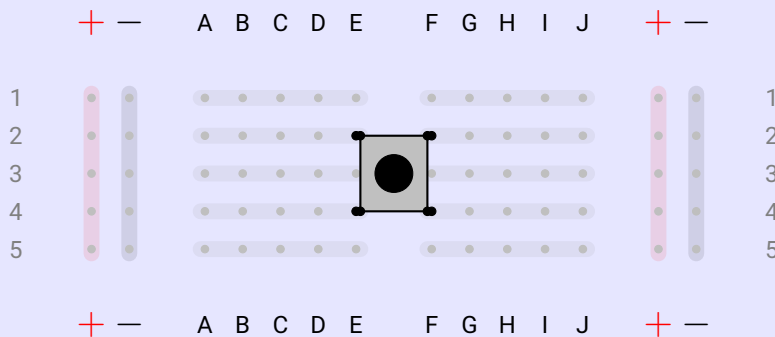
HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

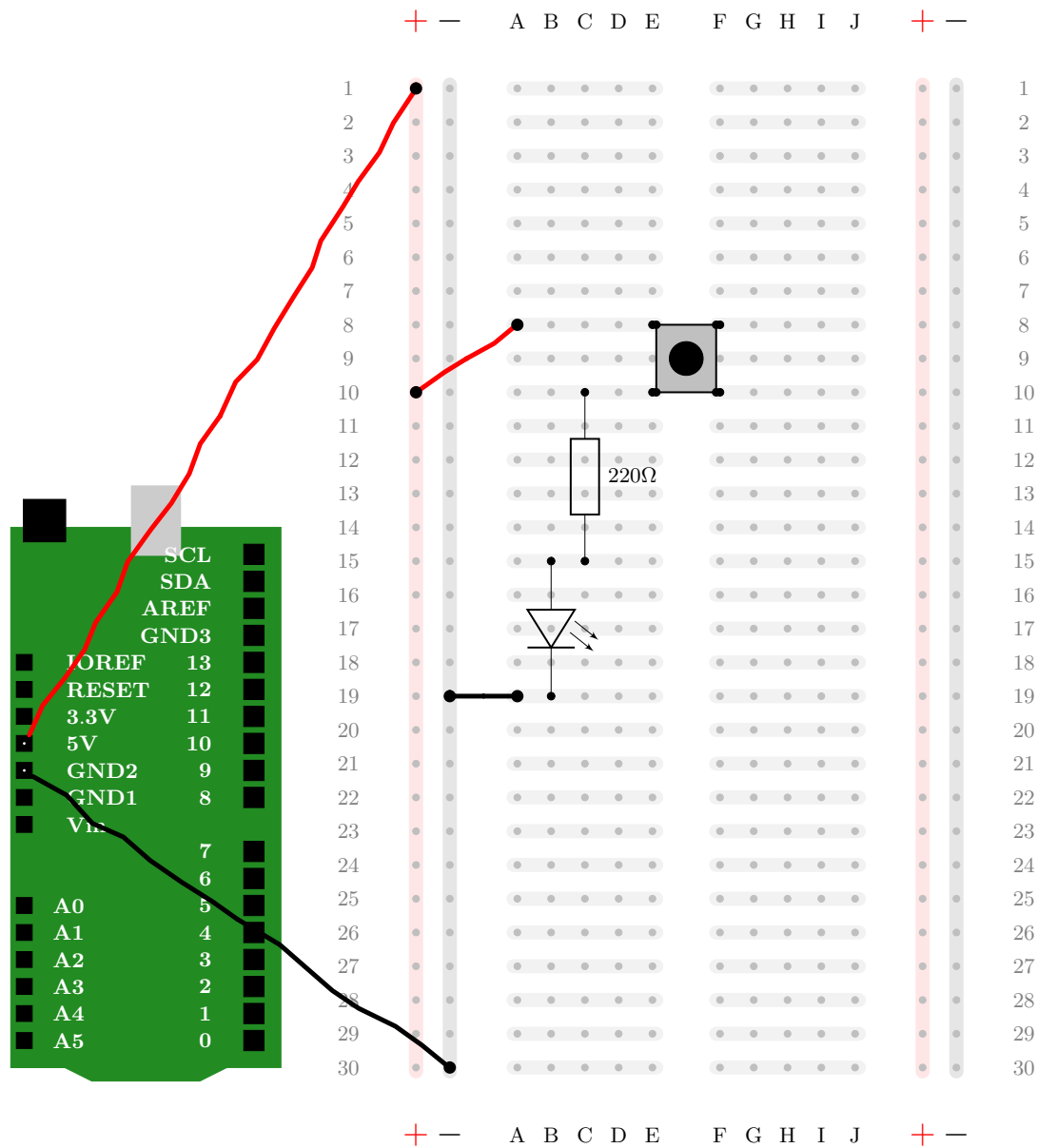
Nyt lisätään piiriin painonappi.

Painonapin kytkeminen

Arduinon mukana tulevassa painonapissa on neljä pinniä, ja asetetaan keskellä olevan uran ylitse. Jos nappia ei paineta, niin A-E ja F-J rivillä 2 ovat samaa pistettä, mutta eri pistettä kuin rivin 4 (A-E) ja (F-J). Jos nappia painetaan, niin molempien rivien (2 ja 4) kaikki pisteet ovat kiinni toisissaan.



Siirretään lyhyempää punaista johtoa kohdasta A10 kohtaan A8. Lisätään painonappi pisteisiin E8, E10, F8 ja F10.



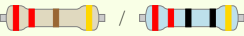
Nyt kytke Arduino takaisin kiinni koneeseen, ja saat LEDin syttymään painamalla nappulaa.

2. Liikennevalot

Lähdetään rakentamaan risteykseen ohjattuja liikennevaloja. Tässä osiossa aloita tyhjästä koekyt-kentälevystä.

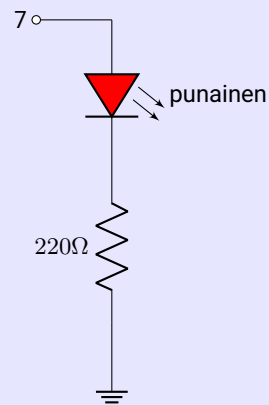
2.1 Ohjelmoitava valo

Tarvikkeet

- Punainen LED
- Vastus 220Ω : 
- Hyppylankoja
- Arduino + koekyt-kentälevy

Piirin toiminta

Ensimmäinen ohjelmoitava piiri. Kirjoite-taan koodi, joka sytyttää ja sammuttaa va-lon koodin mukaisesti.



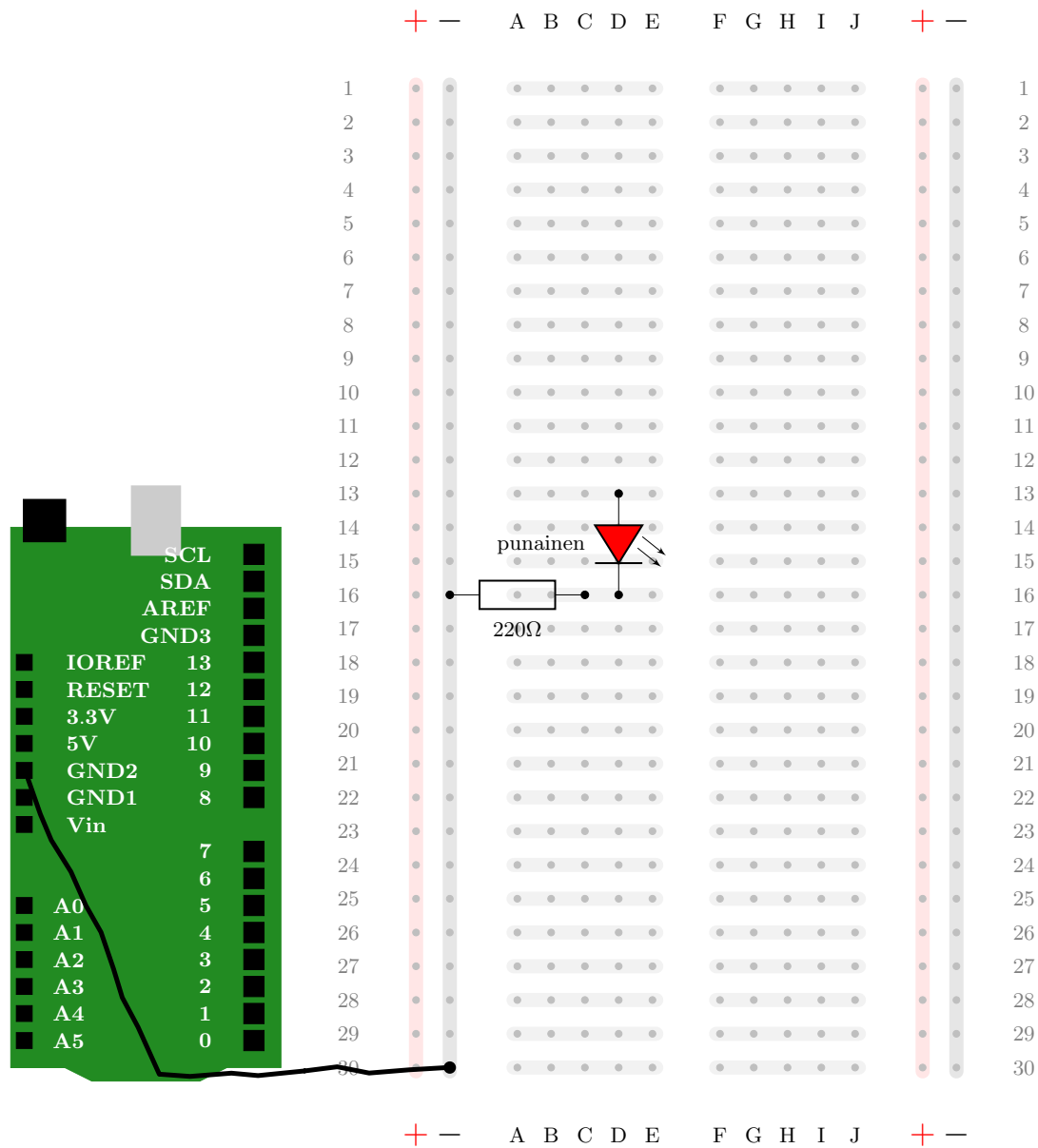
HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

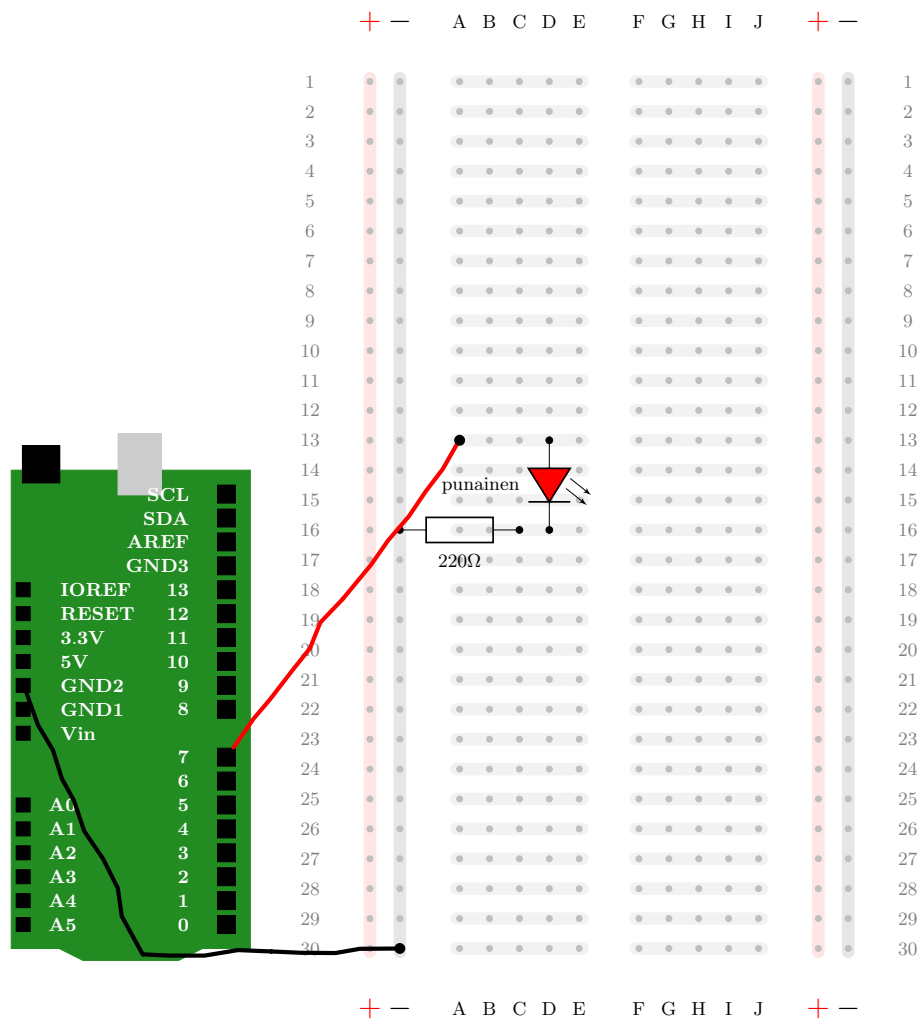
Muista tarkistaa miten päin LEDit kytketään! Tämä löytyy esimerkiksi sivulta 70.

Aiemmin käytimme painonappia valon ohjaamiseen, ja painamalla nappia saimme valon sytty-mään. Nyt tehdään kytkentä, jonka avulla saadaan valo automaattisesti sammumaan ja syttymään.

Kytetään ensin punainen LED ja vastus koekyt-kentälevylle. Ja kytketään myös – maahan.



Aiemmassa kytkennässä rivi 13 kytkettiin suoraan vakiojännitelähteeseen, 5V, mutta nyt koska haluamme ohjelmoida, kytketään rivi 13 käyttämältämme puolelta ohjelmoitavaan pinniin 7 Arduinos-
sa.



Avataan nyt Arduino-sovellus, ja kopioidaan alla oleva koodi sinne.

```
void setup() {
  pinMode(7, OUTPUT); // Punainen LED on pinnissa 7
}

void loop() {
  digitalWrite(7, HIGH); // Laitetaan valo paalle
  delay(1000); // Odotetaan
  digitalWrite(7, LOW); // Laitetaan valo pois paalta
  delay(1000); // Odotetaan
}
```

Käydään vielä koodi lävitse:

Rivit 1-3 ovat alustamista (asetukset) varten. Rivillä 3 kerrotaan mitä haluamme tehdä pinnillä 7:

haluamme ohjata tällä valoa, joten tyyppi on OUTPUT.

Rivit 5-10 ovat miten haluamme ohjata valoa. Kirjoittamalla rivin 6 koodin, asetamme pinnin 7 arvoksi HIGH, eli korkea, eli 5V, ja kuten tiedämme aiemmasta piiristämme, tällöin valo palaa. Sitten seuraavalla rivillä odotamme 1000ms (millisekuntia, eli yhden sekunnin), kunnes teemme rivin 8 operaation: muutamme pinnin 7 arvoksi LOW, eli matala, eli 0V, ja sammutamme valon. Tämän jälkeen odotamme taas yhden sekunnin. Koska koodi on kirjoitettu silmukkaan (loop), toistamme tätä, kunnes kytkemme piirin irti jännitelähteestä.

Nyt voit ladata Arduinolle koodin ja katsoa mitä tapahtuu! Voit myös testata odotusajan muuttamista muuttamalla lukuja 1000 koodissa. Muista, että kyseessä on millisekunnit, joten jos haluat, että valo on päällä 3s ja pois päältä 2s, ensimmäinen 1000 pitää muuttaa luvuksi 3000 ja toinen luvuksi 2000.

Vinkkejä Arduinolla koodaamiseen!

Seuraavassa esitellään Arduinolla koodaamiseen liittyviä vinkkejä.

```
1 // Kaksi kauttaviivaa aloittaa kommentin. Tata ei siis lueta
   koodissa vaan on tiedoksi koodin lukijalle
2
3 pinMode(luku, tyyppi) // Talla voit saataa pinnien 0-13 tekoa
   laittamalla sanan luku tilalle luvun jolta riviltä haluat
   komentaa
4 // Tyyppin tilalle voit kirjoittaa joko OUTPUT eli Arduino lähettää
   signaalia
5 // tai INPUT jolloin Arduino vastaanottaa signaalin
6
7 digitalWrite(luku, arvo) // Talla voit muuttaa pinniin luku
   kirjoitettua arvoa
8 // arvo HIGH vastaa 5V jännitettä
9 // arvo LOW vastaa 0V jännitettä
10
11 delay(luku) // Odottaa luvun verran millisekunteja
```

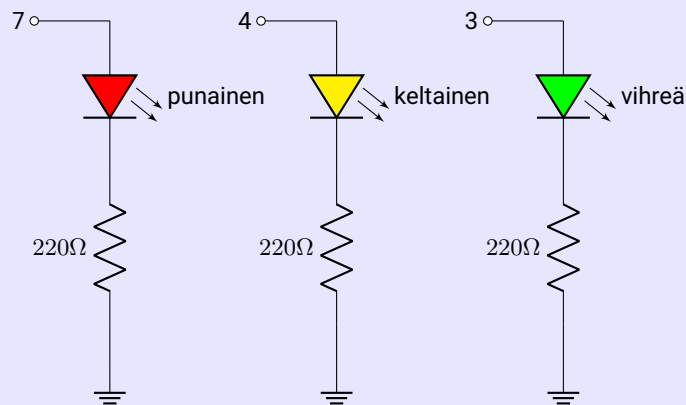
2.2 Ohjelmoitava liikennevalo

Tarvikkeet

- Edellisen kohdan piiri rakennettuna
- Keltainen LED ja vihreä LED
- 2 kpl 220Ω : 
- Hyppylankoja

Piirin toiminta

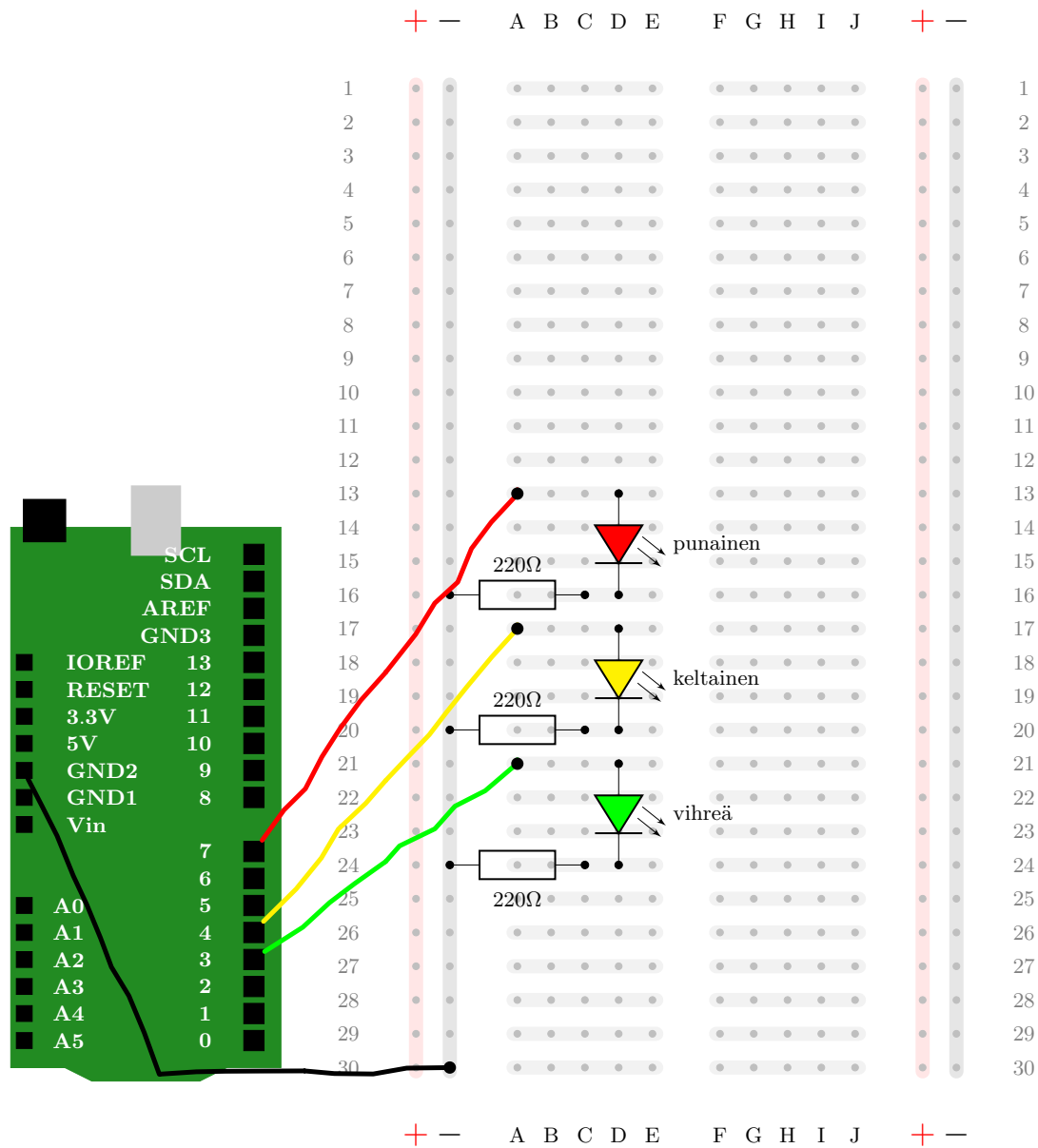
Kirjoitetaan koodi, jonka avulla saadaan liikennevalot toimimaan.



HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

Lisätään ensin keltainen ja vihreä LED ja niiden kanssa sarjassa olevat vastukset ja kytetään keltainen LED pinniin 4 ja vihreä LED pinniin 3. Huomaa, että hyppylankojen värillä ei ole väliä. Selvyiden vuoksi kuvassa on käytetty samaa väriä kuin mihin LEDiin ollaan yhdistämässä.



Nyt lisätään keltaiselle ja vihreällä LEDille (vastaavasti kuin punaisella LEDillä jo on), tieto siitä mitä pinneillä 4 ja 3 halutaan tehdä.

```
void setup() {  
  pinMode(7, OUTPUT); // Punainen LED on pinnissa 7  
  pinMode(4, OUTPUT); // Keltainen LED on pinnissa 4  
  pinMode(3, OUTPUT); // Vihrea LED on pinnissa 2  
}  
  
void loop() {  
  digitalWrite(7, HIGH); // Laitetaan punainen valo paalle  
  digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle  
  digitalWrite(3, HIGH); // Laitetaan vihrea valo paalle  
  delay(1000); // Odotetaan  
  digitalWrite(7, LOW); // Laitetaan punainen valo pois paalta  
  digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta  
  digitalWrite(3, LOW); // Laitetaan vihrea valo pois paalta  
  delay(1000); // Odotetaan  
}
```

Nyt yllä oleva koodi vilkuttaa kaikkea kolmea valoa yhtä aikaa. Vaihda rivien järjestystä, ja lisää tarvittaessa odotuskomentoja tai päälle/pois komentoja, jotta saat valot ohjelmoitua.

Koodaa!

Miten muutat koodia, jotta saat liikennevalot toimimaan kuten oikeat valot toimivat?

```
1 void setup() {  
2   pinMode(7, OUTPUT); // Punainen LED on pinnissa 7  
3   pinMode(4, OUTPUT); // Keltainen LED on pinnissa 4  
4   pinMode(3, OUTPUT); // Vihrea LED on pinnissa 2  
5 }  
6  
7 void loop() {  
8   digitalWrite(7, HIGH); // Laitetaan punainen valo paalle  
9   delay(1000); // Odotetaan  
10  digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle  
11  delay(1000); // Odotetaan  
12  digitalWrite(7, LOW); // Laitetaan punainen valo pois paalta  
13  digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta  
14  digitalWrite(3, HIGH); // Laitetaan vihrea valo paalle  
15  delay(1000); // Odotetaan  
16  digitalWrite(3, LOW); // Laitetaan vihrea valo pois paalta  
17  digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle  
18  delay(1000);  
19  digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta  
20  digitalWrite(7, HIGH); // Laitetaan punainen valo paalle  
21  delay(1000); // Odotetaan  
22 }
```

Kyseessä on esimerkkikoodi, ja tämän jälkeen voidaan puhua siitä, kauanko kukin valo palaa ja tarvitseeko viimeisenä olla punainen valo päälle vai ei.

2.3 Painonapin lisääminen

Tarvikkeet

- Edellisen kohdan piiri rakennettuna
- Painonappi
- Vastus 220Ω:  / 
- Hyppylankoja

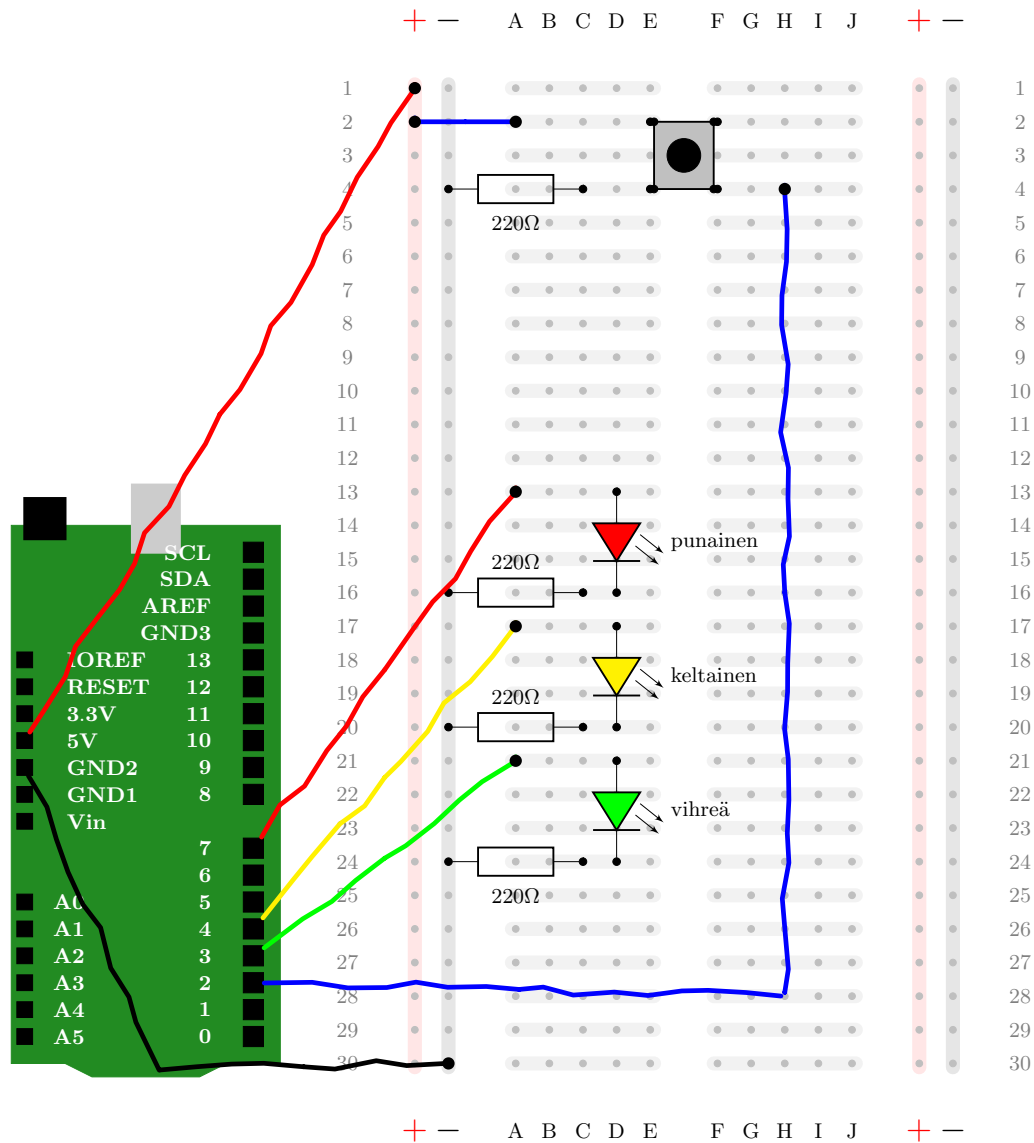
Piirin toiminta

Lisätään painonappi, jota painamalla saadaan liikennevalojen toiminta muuttumaan.

HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

Käytimme jo aiemmin painonappia manuaalisesti valon ohjaamiseen, mutta lisätään tämä nyt koodimme. Lisää aiempaan kytkentääsi painonappi sekä vastus ja kolme hyppylankaa (kuvassa punainen ja kaksi sinistä).



Nyt muokataan koodia ottamaan huomioon painonappi. Tämä esimerkkikoodi sytyttää kaikki valot kun nappia painetaan ja ne sammuvat kun nappia ei paineta.

```
void setup() {  
  pinMode(7, OUTPUT); // Punainen LED on pinnissa 7  
  pinMode(4, OUTPUT); // Keltainen LED on pinnissa 4  
  pinMode(3, OUTPUT); // Vihrea LED on pinnissa 2  
  pinMode(2, INPUT); // Lisataan painonappi tuloksi  
}  
  
void loop() {  
  bool buttonState = digitalRead(2); // Luetaan painonapin tila  
  
  // Tarkastetaan onko nappi painettu  
  if (buttonState == HIGH) {  
    // Jos on laitetaan LEDit paalle  
    digitalWrite(7, HIGH); // Laitetaan punainen valo paalle  
    digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle  
    digitalWrite(3, HIGH); // Laitetaan vihrea valo paalle  
  } else {  
    // Jos ei, niin laitetaan LEDit pois paalta  
    digitalWrite(7, LOW); // Laitetaan punainen valo pois paalta  
    digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta  
    digitalWrite(3, LOW); // Laitetaan vihrea valo pois paalta  
  }  
}
```

Vinkkejä Arduinolla koodaamiseen!

```
1 muuttuja = digitalRead(luku); // Lukee pinnin luku arvon ja
    tallentaa sen muuttujaan muuttuja
2 // Huomaa etta muuttujalla pitaa olla annettuna tyyppi
3 // Esimerkiksi bool:
4 bool muuttuja = digitalRead(luku);
5 // bool voi olla joko HIGH tai LOW sen mukaan mita tassa
    tapauksessa luettiin piirista
6 // bool on kateva tilanteissa joissa arvo on joko HIGH tai LOW
7
8 // Ehtorakenne
9 // Tarkastetaan muuttujan arvo.
10 if (muuttuja == HIGH) {
11     // jos muuttujan arvo on HIGH tehdään koodi tassa valissa
12 } else {
13     // muuten (eli kun LOW) tehdään mita tassa kirjoitetaan
14 }
```

Haaste!

Miten muutat koodia, jotta kun nappia painetaan, vilkkuu vain keltainen valo ja kun nappia ei paineta, liikennevalot toimivat normaalisti?

Koodaa!

```
1 void setup() {
2   pinMode(7, OUTPUT); // Punainen LED on pinnissa 7
3   pinMode(4, OUTPUT); // Keltainen LED on pinnissa 4
4   pinMode(3, OUTPUT); // Vihrea LED on pinnissa 2
5   pinMode(2, INPUT); // Lisataan painonappi tuloksi
6 }
7
8 void loop() {
9   bool buttonState = digitalRead(2); // Luetaan painonapin tila
10  // Tarkastetaan onko nappi painettu
11  if (buttonState == HIGH) {
12    // Jos on laitetaan LEDit paalle
13    digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
14    digitalWrite(7, LOW); // Sammutetaan varmuuden vuoksi punainen
        valo
```

```
15     digitalWrite(3, LOW); // Sammutetaan varmuuden vuoksi vihrea
    valo
16     delay(1000); // Odotetaan
17     digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta
18     delay(1000);
19 } else {
20     digitalWrite(7, HIGH); // Laitetaan punainen valo paalle
21     delay(1000); // Odotetaan
22     digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
23     delay(1000); // Odotetaan
24     digitalWrite(7, LOW); // Laitetaan punainen valo pois paalta
25     digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta
26     digitalWrite(3, HIGH); // Laitetaan vihrea valo paalle
27     delay(1000); // Odotetaan
28     digitalWrite(3, LOW); // Laitetaan vihrea valo pois paalta
29     digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
30     delay(1000);
31     digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta
32     digitalWrite(7, HIGH); // Laitetaan punainen valo paalle
33     delay(1000); // Odotetaan
34 }
35 }
```

Nyt huomionarvoista on se, koska piiri reagoi napinpainallukseen? Yllä olevalla koodilla reagointi tapahtuu vasta kun valo on vaihdettu punaiseksi ja valot käynnistyvät taas punaisesta, kun napista päästetään irti.

Haaste!

Laita nappi toimimaan valokatkaisijan tavoin. Kun nappia painetaan kerran, alkaa ohjelma vilkuttamaan keltaista valoa ja jos nappia painetaan uudelleen, liikennevalot toimivat taas.

Vinkkejä Arduinolla koodaamiseen!

```
1 bool ledStatus = true; // Luodaan muuttuja, jonka arvo on totta
2
3 ledStatus = !ledStatus; // Muuttaa muuttujan ledStatus arvoksi
    arvon epatosi (jos arvo oli tosi) tai tosi (jos arvo oli
    epatosi)
```

Koodaa!

```
1 // Alustetaan muuttuja
2 bool ledStatus = true;
3
4 void setup() {
5   pinMode(7, OUTPUT); // Punainen LED on pinnissa 7
6   pinMode(4, OUTPUT); // Keltainen LED on pinnissa 4
7   pinMode(3, OUTPUT); // Vihrea LED on pinnissa 2
8   pinMode(2, INPUT); // Lisataan painonappi tuloksi
9 }
10
11 void loop() {
12   bool buttonState = digitalRead(2); // Luetaan painonapin tila
13
14   if (buttonState == HIGH) {
15     ledStatus = !ledStatus
16   }
17
18   // Tarkastetaan mitä tehdään
19   if (ledStatus == HIGH){
20     // Jos on laitetaan LEDit paalle
21     digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
22     digitalWrite(7, LOW); // Sammutetaan varmuuden vuoksi punainen
        valo
23     digitalWrite(3, LOW); // Sammutetaan varmuuden vuoksi vihrea
        valo
24     delay(1000); // Odotetaan
25     digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta
26     delay(1000);
27   } else {
28     digitalWrite(7, HIGH); // Laitetaan punainen valo paalle
29     delay(1000); // Odotetaan
30     digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
31     delay(1000); // Odotetaan
32     digitalWrite(7, LOW); // Laitetaan punainen valo pois paalta
33     digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta
34     digitalWrite(3, HIGH); // Laitetaan vihrea valo paalle
35     delay(1000); // Odotetaan
36     digitalWrite(3, LOW); // Laitetaan vihrea valo pois paalta
37     digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
```

```
38     delay(1000);  
39     digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta  
40     digitalWrite(7, HIGH); // Laitetaan punainen valo paalle  
41     delay(1000); // Odotetaan  
42 }  
43 }
```

Nyt huomattavaa on se, että napin painallus pitää ajoittaa oikein, sillä väärässä vaiheessa napin painallusta ei huomata, eli nappia on painettava pitkään, jotta muutos välittyy koodille.

Haaste!

Muuta koodin toimintaa niin, että edellinen koodi reagoi heti kun nappia painetaan.

Vinkkejä Arduinolla koodaamiseen!

```
1 // Seuraava rivi lisataan setup() funktioon
2 attachInterrupt(digitalPinToInterrupt(luku), keskeytys, FALLING);
3 // attachInterrupt on komento jolla luodaan keskeytys
4 // digitalPinToInterrupt(luku) kertoo mistä pinnista saadaan
  painallus
5 // keskeytys on aliohjelman nimi, johon siirrytään
6 // FALLING on parametri joka tarkoittaa kun nappia painetaan alas
7 // Vaihtoehtona ovat myös
8 // LOW kun pinnin luku jännite on 0V
9 // CHANGE kun pinnin luku jännite muuttuu
10 // RISING kun pinnin tila muuttuu nolasta viiteen volttiin
11 // FALLING kun pinnin tila muuttuu viidesta voltista noltaan
12
13 // Maaritetaan uusi funktio olemassa olevien peraan
14 void keskeytys() {
15   // Mita funktio tekee?
16   // Esimerkiksi vaihtaa muuttujan ledState arvo:
17   ledState = !ledState;
18 }
19 // Talloin ohjelman alussa pitaa olla maaritys
20 volatile bool ledState = false;
21 // volatile tarkoittaa, etta muuttujan arvo tarkastetaan aina sita
  kaytettaessa
22 // muuttujan ledState arvo voi siis muuttua milloin vain ohjelman
  aikana
```

Koodaa!

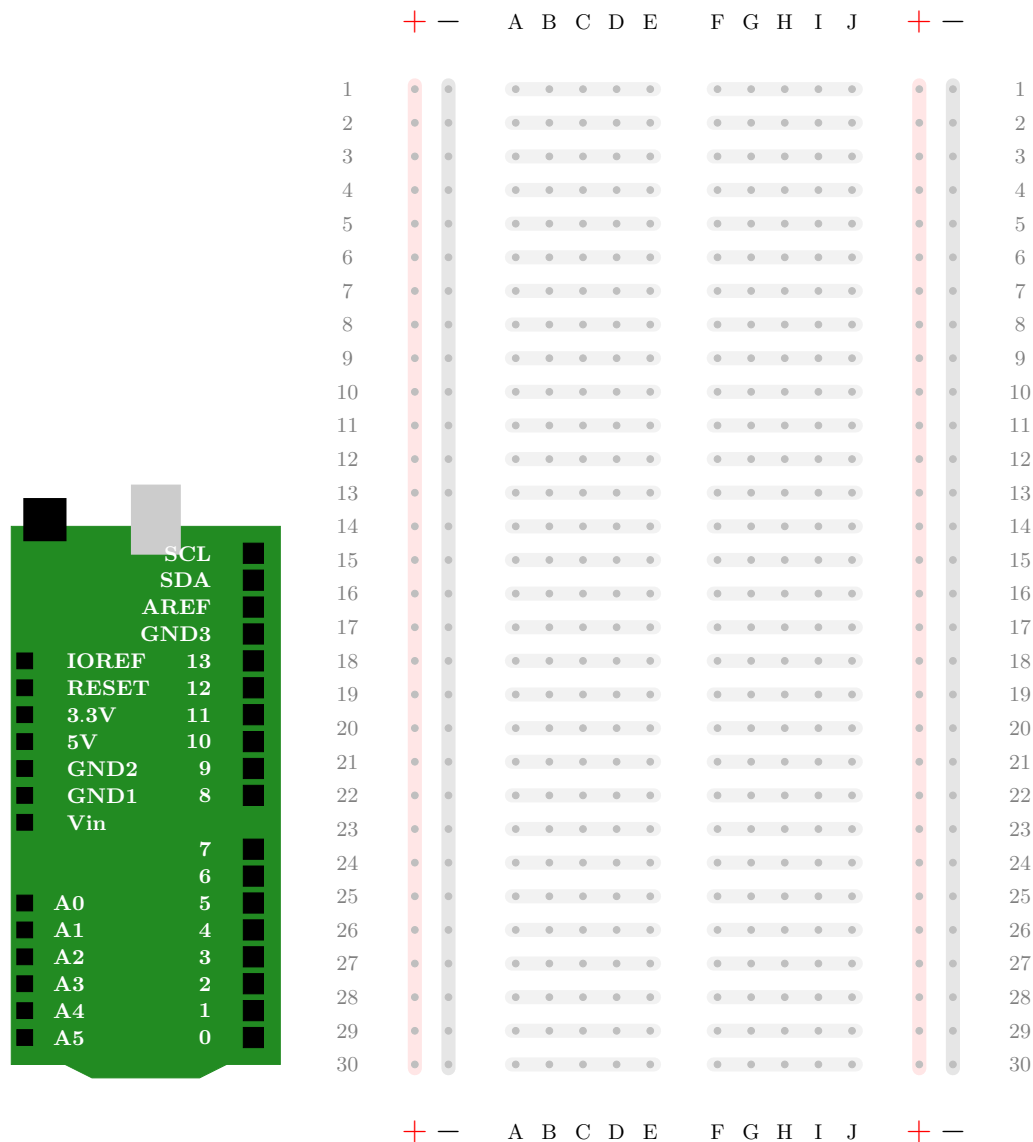
```
1 // Alustetaan muuttuja
2 volatile bool ledStatus = false;
3
4 void setup() {
5   pinMode(7, OUTPUT); // Punainen LED on pinnissa 7
6   pinMode(4, OUTPUT); // Keltainen LED on pinnissa 4
7   pinMode(3, OUTPUT); // Vihrea LED on pinnissa 2
8   pinMode(2, INPUT); // Lisataan painonappi tuloksi
9   attachInterrupt(digitalPinToInterrupt(2), keskeytys, FALLING);
10 }
```

```
11
12 void loop() {
13     // Tarkastetaan onko nappi painettu
14     if (ledStatus == HIGH) {
15         // Jos on laitetaan LEDit paalle
16         digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
17         digitalWrite(7, LOW); // Sammutetaan varmuuden vuoksi punainen
            valo
18         digitalWrite(3, LOW); // Sammutetaan varmuuden vuoksi vihrea
            valo
19         delay(1000); // Odotetaan
20         digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta
21         delay(1000);
22     } else {
23         digitalWrite(7, HIGH); // Laitetaan punainen valo paalle
24         delay(1000); // Odotetaan
25         digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
26         delay(1000); // Odotetaan
27         digitalWrite(7, LOW); // Laitetaan punainen valo pois paalta
28         digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta
29         digitalWrite(3, HIGH); // Laitetaan vihrea valo paalle
30         delay(1000); // Odotetaan
31         digitalWrite(3, LOW); // Laitetaan vihrea valo pois paalta
32         digitalWrite(4, HIGH); // Laitetaan keltainen valo paalle
33         delay(1000);
34         digitalWrite(4, LOW); // Laitetaan keltainen valo pois paalta
35         digitalWrite(7, HIGH); // Laitetaan punainen valo paalle
36         delay(1000); // Odotetaan
37     }
38 }
39
40 void keskeytys() {
41     ledStatus = !ledStatus;
42 }
```

Lisäksi voi miettiä esimerkiksi miten jalankulkijan painonapin saisi mukaan. Huomaa, että tämän työn kanssa on tärkeää painaa painonappia hyvin, Arduino ei aina kunnolla huomaa tai huomaa liian monta painallusta putkeen yhden sijasta. Tätä voi kiertää koodaamalla esimerkiksi ajastimen painallusten välisen ajan havainnointiin, mutta tämä on jätetty ohjeesta pois.

2.4 Suunnittelu omat liikennevalot risteykseen

Nyt kun ohjelmointia on kokeiltu monelta eri kannalta, voit suunnitella kokonaisen risteuksen liikennevalot. Piirrä alle risteyksesi. Lisäksi on tilaa piirtää kuva koekytkentälevystäsi, jotta näet miten olet piirin kytkenyt. Sitä seuraavalla sivulla on tilaa suunnitella koodiasi.



Koodaa!

Tähän tehtävään ei ole malliratkaisua.

3. Lämpötila-anturi ja valoanturi

3.1 Lämpömittari

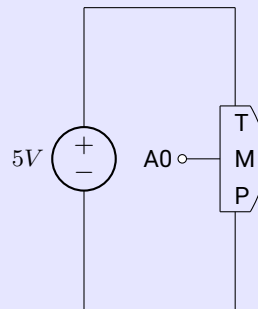
3.1.1 Lämpötila-anturin käyttöönotto

Tarvikkeet

- Lämpötila-anturi (TMP36)
- Koekytkenälevy
- Arduino UNO
- Hyppylankoja

Piirin toiminta

Lämpötila-anturin kytkentä ja käyttö, saadun anturin arvon muuntaminen Celcius-asteiksi.



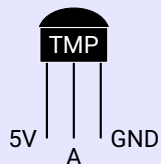
HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

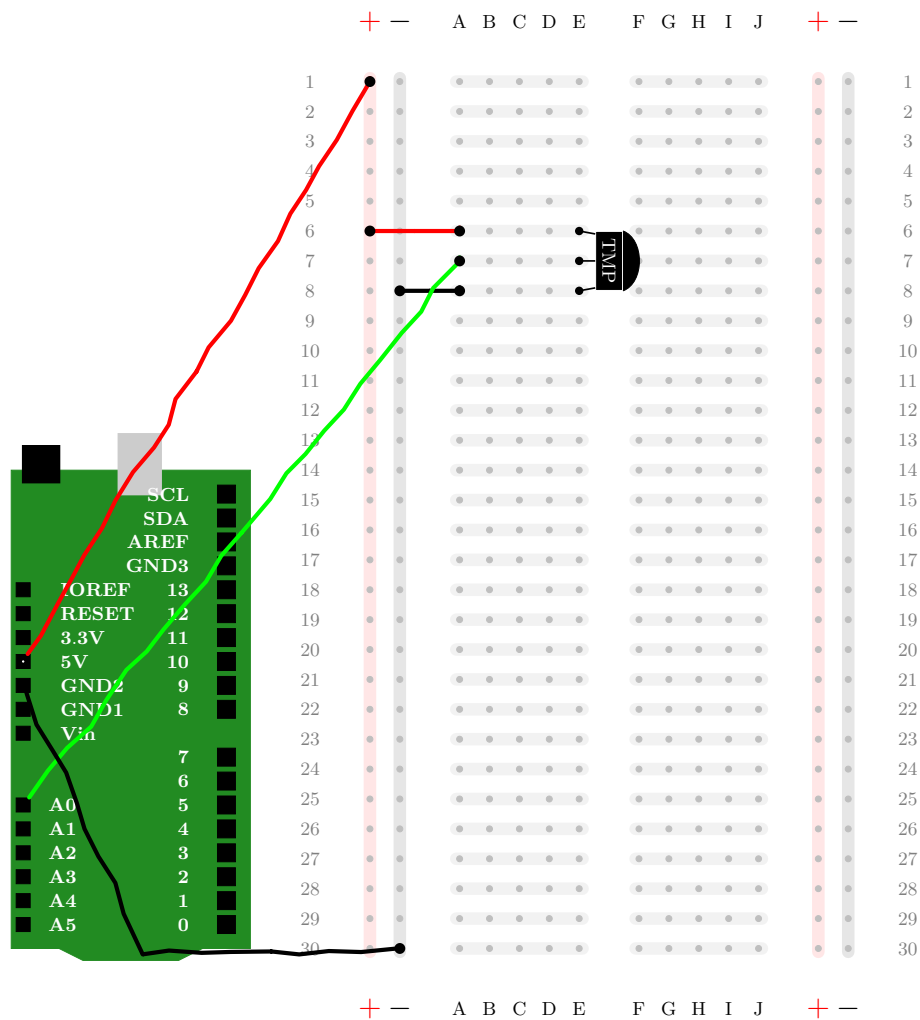
Lämpötila-anturi

Huomaa, että on merkitystä miten päin lämpötila-anturi kytketään piiriin!

Jos tasainen sivu on kohti sinua, silloin vasemman puoleisin jalka on käyttöjännite (5V), keskimmaiselta jalalta (A niin kuin anturi) voidaan lukea tulos ja oikean puoleisin jalka kytketään maahan (GND).



Jos kytket anturin väärinpäin, se voi kuumentua. Ole siis tarkkana!



Vinkkejä Arduinolla koodaamiseen!

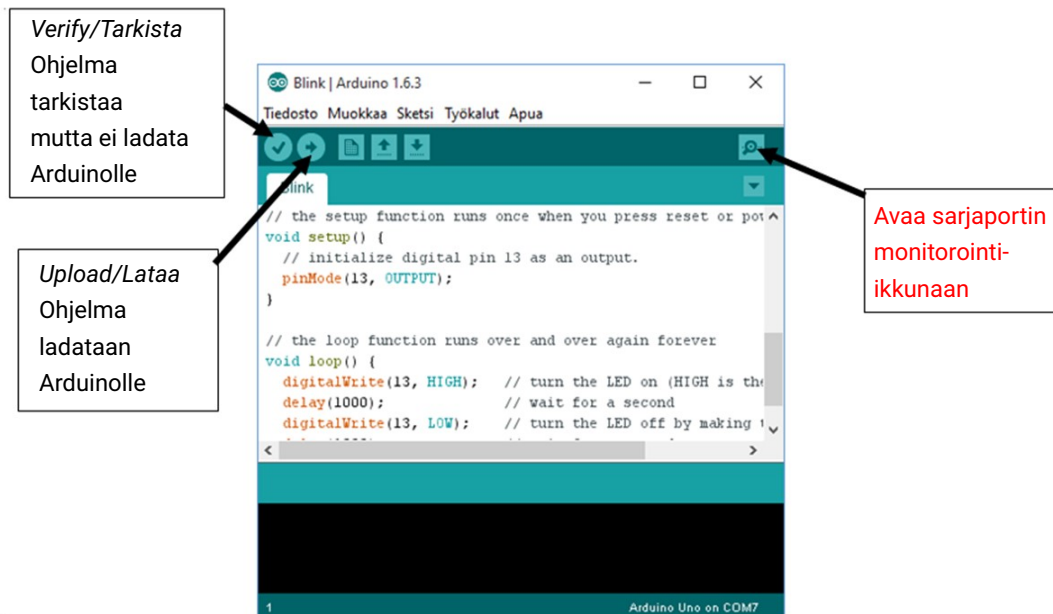
```

1 // Seuraava rivi kuuluu setup()-funktion sisälle
2 Serial.begin(9600); // Aloittaa kommunikoinnin tietokoneen kanssa
3 // 9600 kertoo kuinka monta bittia sekunnissa lahetetaan.
4
5 // Kommentoja, joilla voidaan tulostaa
6 Serial.print(muuttuja); // Tulostetaan muuttujan muuttuja arvo
7 Serial.print("Teksti:"); // Tulostetaan Teksti:
8 Serial.println(muuttuja); // Sama kuin muuttujan tulostus, mutta
   rivinvaihto lopussa
9 Serial.print("\n"); // Rivinvaihto manuaalisesti

```

Nämä tulostetut asiat löytyvät Arduinon "Serial Monitor":sta, eli oikean yläkulman suurennuslasi-

kuvakkeen takaa.



Kopioi alla oleva koodi Arduinoon.

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int sensorVal = analogRead(A0); // Luetaan anturin arvo portista A0
  Serial.print("Anturin arvo: "); // lisataan selitys luvulle
  Serial.print(sensorVal); // ja tulostetaan anturin arvo
  float voltage = (sensorVal/1024.0)*5.0; // Muunna anturin arvo
  volteiksi
  Serial.print(". Voltteina: "); // Selitys
  Serial.print(voltage); // Ja tulosta se
  float temp = (voltage-0.5)*100; // Muunnetaan jännite Celcius-
  asteiksi
  Serial.print(". Astetta "); // Selitys
  Serial.print(temp); // ja tulostetaan lampotila
  Serial.print(" C\n"); // Celcius-astetta ja vaihdetaan rivia
  delay(1000); // Lisataan odotus, jotta ehditaan lukea tulos
}
```

Lämpötila

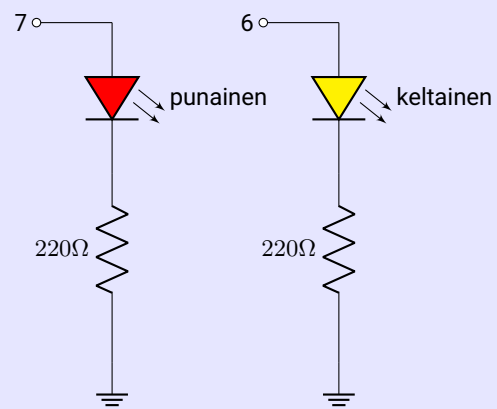
Kirjoita ylös saamasi huoneen lämpötila. Esimerkiksi 20.0 astetta.

3.1.2 LEDien lisääminen**Tarvikkeet**

- Edellisen kohdan piiri rakennettuna
- Keltainen ja punainen LED
- 2 kappaletta 220Ω vastuksia:

**Piirin toiminta**

Lisätään havainnointia varten kaksi LEDiä, joiden avulla saadaan tietää onko huoneessa sopivan lämpöistä, kuumaa vai liian kuumaa.

**HUOM!**

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

Muista tarkistaa miten päin LEDit kytketään! Tämä löytyy esimerkiksi sivulta 70.

Vinkkejä Arduinolla koodaamiseen!

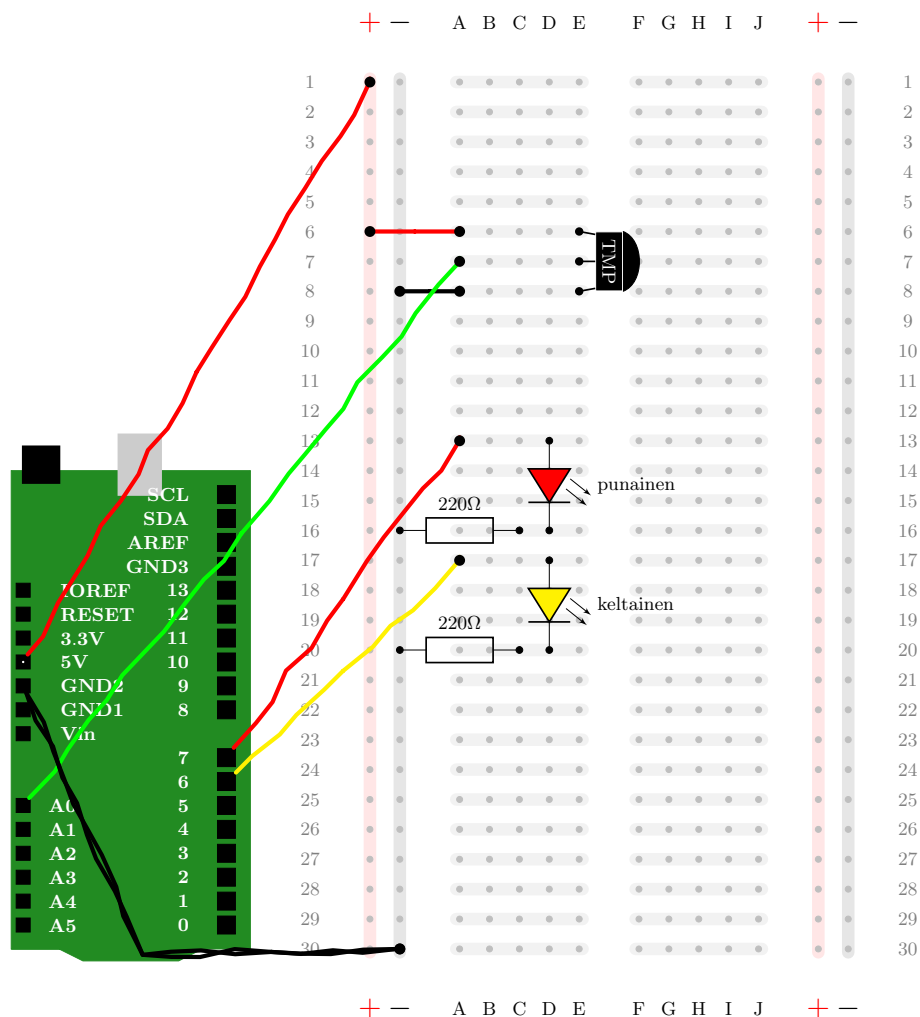
```
1 // Halutaan tehdä monta asiaa riippuen monesta asiasta
2 if (ehto) {
3     // Mita tehdään jos ehto patee
4 } else if (ehto2) {
5     // Mita tehdään jos ehto2 patee
6 } else if (ehto3) {
7     // Naita voit lisata niin monta kuin tarvitset
8 } else {
9     // Ja halutessa voit maaritella myös mita tehdään muulloin
```

```

10 }
11
12 // Erilaisia ehtoja
13 muuttuja < luku // Muuttujan muuttuja arvo on pienempi kuin
    annettu luku
14 muuttuja <= luku // Muuttujan muuttuja arvo on pienempi tai yhtä
    suuri kuin luku
15 muuttuja > luku // Muuttuja on suurempi kuin luku
16 // Kaksi ehtoa yhtä aikaa
17 muuttuja >= luku1 && muuttuja < luku2 // Muuttuja on suurempi tai
    yhtasuuri kuin luku1 JA pienempi kuin luku2

```

Lisätään LEDit sekä vastukset kuvan mukaisesti.



Lisätään koodiin LEDien ohjaaminen.

```
// Lisataan saatu huoneenlamppu tahan
const float huoneLamp = 20.0; // Muokkaa lukua edellisen tehtävän
    mukaiseksi

void setup() {
    Serial.begin(9600);
    pinMode(7, OUTPUT); // Punainen LED on portissa 7
    pinMode(6, OUTPUT); // Keltainen LED on portissa 6
}

void loop() {
    int sensorVal = analogRead(A0); // Luetaan anturin arvo portista A0
    Serial.print("Anturin arvo: "); // lisataan selitys luvulle
    Serial.print(sensorVal); // ja tulostetaan anturin arvo
    float voltage = (sensorVal/1024.0)*5.0; // Muunna anturin arvo
        volteiksi
    Serial.print(". Voltteina: "); // Selitys
    Serial.print(voltage); // Ja tulosta se
    float temp = (voltage-0.5)*100; // Muunnetaan jännite Celcius-
        asteiksi
    Serial.print(". Astetta "); // Selitys
    Serial.print(temp); // ja tulostetaan lampotila
    Serial.print(" C\n"); // Celcius-astetta ja vaihdetaan rivia

    // Lisataan mita tehdään jos lampotila nousee tai laskee liikaa
    if (temp <= huoneLamp){
        // Jos huoneen lamppu: sytytetään keltainen LED
        digitalWrite(7, LOW); // Sammutetaan punainen LED tarvittaessa
        digitalWrite(6, HIGH); // Sytytetään keltainen LED
    }else if(temp>huoneLamp+2 && temp <huoneLamp+4){
        // Lamppu on nousemassa: sytytetään molemmat valot
        digitalWrite(7, HIGH); // Sytytetään punainen valo
        digitalWrite(6, HIGH); // Sytytetään keltainen valo
    }else if (temp>huoneLamp+4) {
        // Lamppu on noussut liikaa, varoitetaan pelkällä punaisella valolla
        digitalWrite(7, HIGH); // Sytytetään punainen valo
        digitalWrite(6, LOW); // Sammutetaan keltainen valo
    }
    delay(1000); // Lisataan odotus, etta ehditaan lukea teksti
}
```

Nyt, avaa jälleen sarjaportin monitorointi-ikkuna. Ota kevyesti kiinni lämpötila-anturista ja katso missä vaiheessa punainen LED syttyy. Saatko myös keltaisen LEDin sammumaan?

Haaste!

Muuta koodia niin, että jos huoneessa on vähintään 2 astetta kylmempää kuin normaalisti, keltainen valo palaa. Jos huoneessa on taas normaalin lämpöistä, mikään valo ei pala. Jos huoneessa on vähintään 2 astetta lämpimämpää kuin normaalisti, punainen valo palaa.

Koodaa!

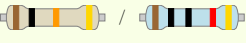
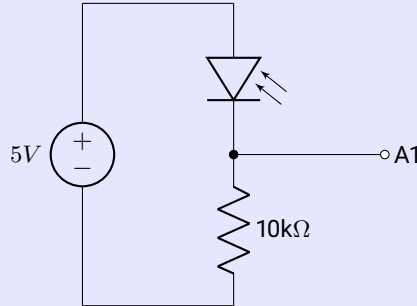
```
1 // Lisataan saatu huoneenlamppu tahan
2 const float huoneLamp = 20.0; // Muokkaa lukua todelliseksi
   lampotilaksi
3
4 void setup() {
5   Serial.begin(9600);
6   pinMode(7, OUTPUT); // Punainen LED on portissa 7
7   pinMode(6, OUTPUT); // Keltainen LED on portissa 6
8 }
9
10 void loop() {
11   int sensorVal = analogRead(A0); // Luetaan anturin arvo portista
      A0
12   Serial.print("Anturin arvo: "); // lisataan selitys luvulle
13   Serial.print(sensorVal); // ja tulostetaan anturin arvo
14   float voltage = (sensorVal/1024.0)*5.0; // Muunna anturin arvo
      volteiksi
15   Serial.print(". Voltteina: "); // Selitys
16   Serial.print(voltage); // Ja tulosta se
17   float temp = (voltage-0.5)*100; // Muunnetaan jännite Celcius-
      asteiksi
18   Serial.print(". Astetta "); // Selitys
19   Serial.print(temp); // ja tulostetaan lampotila
20   Serial.print(" C\n"); // Celcius-astetta ja vaihdetaan rivia
21
22   // Lisataan mitä tehdään jos lampotin
23   if (temp < huoneLamp-2){
24     // Huoneessa alkaa olla kylmä
25     digitalWrite(7, LOW); // Sammutetaan punainen LED tarvittaessa
26     digitalWrite(6, HIGH); // Sytytetään keltainen LED
27   }else if(temp >= huoneLamp-2 && temp < huoneLamp+2){
```

```
28 // Normaalihuoneen lamppu
29 digitalWrite(7, LOW); // Sytytetään punainen valo
30 digitalWrite(6, LOW); // Sytytetään keltainen valo
31 }else if (temp>huoneLamp+2) {
32 // Lamppu on noussut liikaa, varoitetaan pelkällä punaisella
    valolla
33 digitalWrite(7, HIGH); // Sytytetään punainen valo
34 digitalWrite(6, LOW); // Sammutetaan keltainen valo
35 }
36 delay(1000); // Odotetaan, että ehditaan lukea
37 }
```

Kylmemmäksi muuttumista voi olla vaikeampi havaita, varsinkin jos esimerkiksi ikkunaa ei saa auki, tai ei voida jäähdyttää sormia kylmällä vedellä. Muista, että kädet pitää kuitenkin olla kuivat, jotta voidaan koskea lämpöanturiin!

3.2 Valoanturi

3.2.1 Valoanturin käyttöönotto

Tarvikkeet	Piirin toiminta
• Valodiodi • Koekytkenälevy • Arduino UNO • Hyppylankoja • Vastus 10kΩ: 	Valodiodin käyttö. 

HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

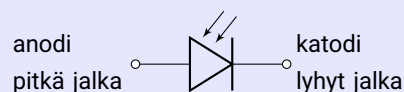
Voit jättää vielä edellisen kohdan piirin purkamatta! Sitä tarvitaan seuraavassa työssä. Tämän työn pystyy rakentamaan myös ilman edellisen kohdan piiriä, mutta seuraavassa työssä on käytössä sekä valo- että lämpöanturit.

Valoanturin kytkeminen

Huomaa, että on merkitystä kummin päin valoanturi kytketään koekytkenälevylle! Valoanturissa on kaksi jalkaa, joista toinen on pidempi. Valodiodi näyttää kirkkaalta LEDiltä, mutta siitä puuttuu pyörästetty yläosa, eli yläpinta on tasainen (toisin kuin LEDeillä, joilla se on pyöreä). Piirrosmerkissä nuolet tulevat kohti diodia, eli olemme ottamassa valoa sisään (emme lähettämässä sitä ulos kuten LEDin kanssa).

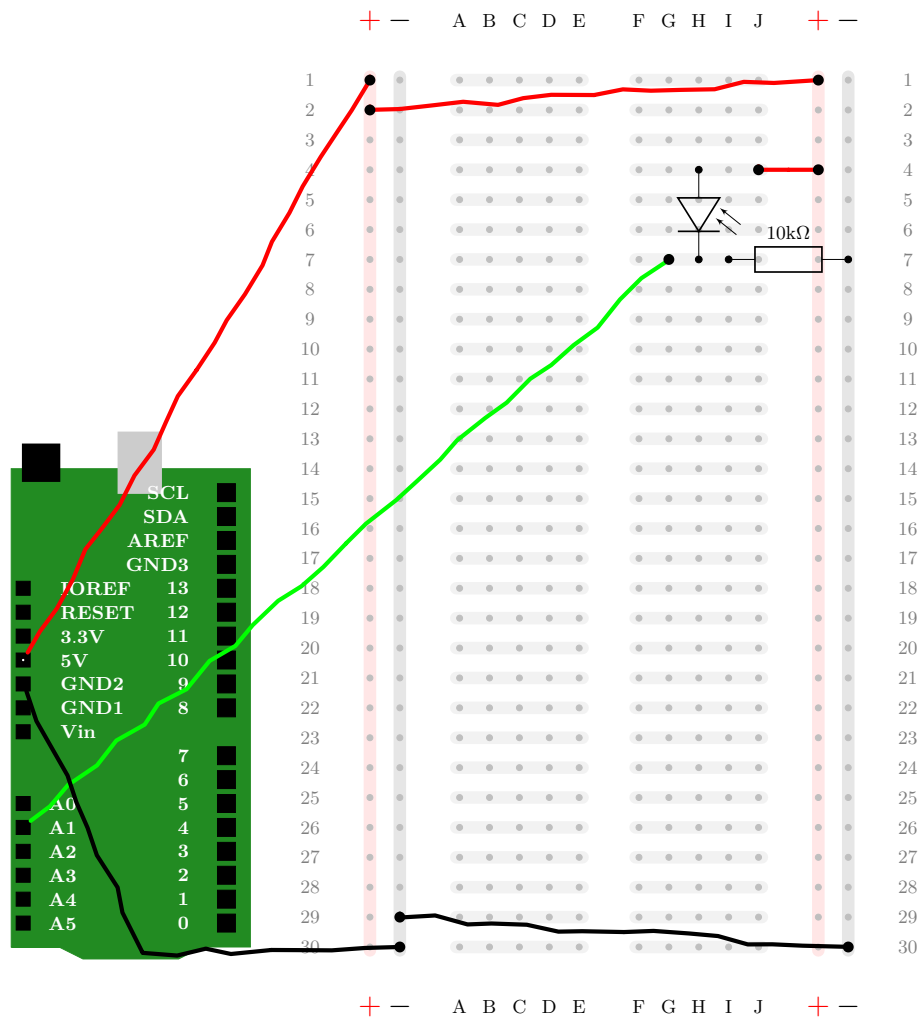
Valoanturin muovi on toiselta puolelta kaareva, ja toiselta siinä on suora leikkaus, lyhyempi jalka on suoran leikkauksen puolella. Lyhyemmän jalan nimi on katodi ja pidemmän jalan anodi.

Piirrosmerkissä:



Valodiodi kytketään myös aina vastuksen kanssa sarjaan, nyt käytössä on 10kΩ vastus.

Huomaa, että lisäsimme käyttöön myös oikealle puolelle käyttöjännitteen ja maan.



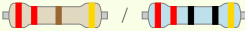
```
void setup() {
    Serial.begin(9600);
}

void loop() {
    int valoa = analogRead(A1); // Luetaan anturin arvo portista A1
    Serial.print("Anturin arvo: "); // Selitys
    Serial.println(valoa); // ja tulostetaan se
    delay(1000); // Odotetaan, että ehditaan lukea
}
```

Valoanturin arvo

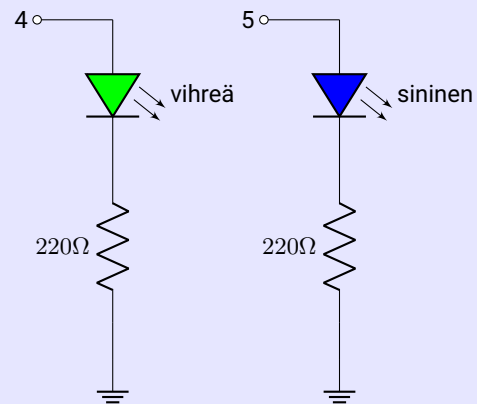
Kirjoita ylös saamasi huoneen valoisuutta kuvaava arvo. Esimerkiksi 400.

3.2.2 LEDien lisääminen**Tarvikkeet**

- Edellisen kohdan piiri rakennettuna
- Sininen ja vihreä LED
- 2 kappaletta 220Ω vastuksia:

- Taskulamppu tai muu valonlähde

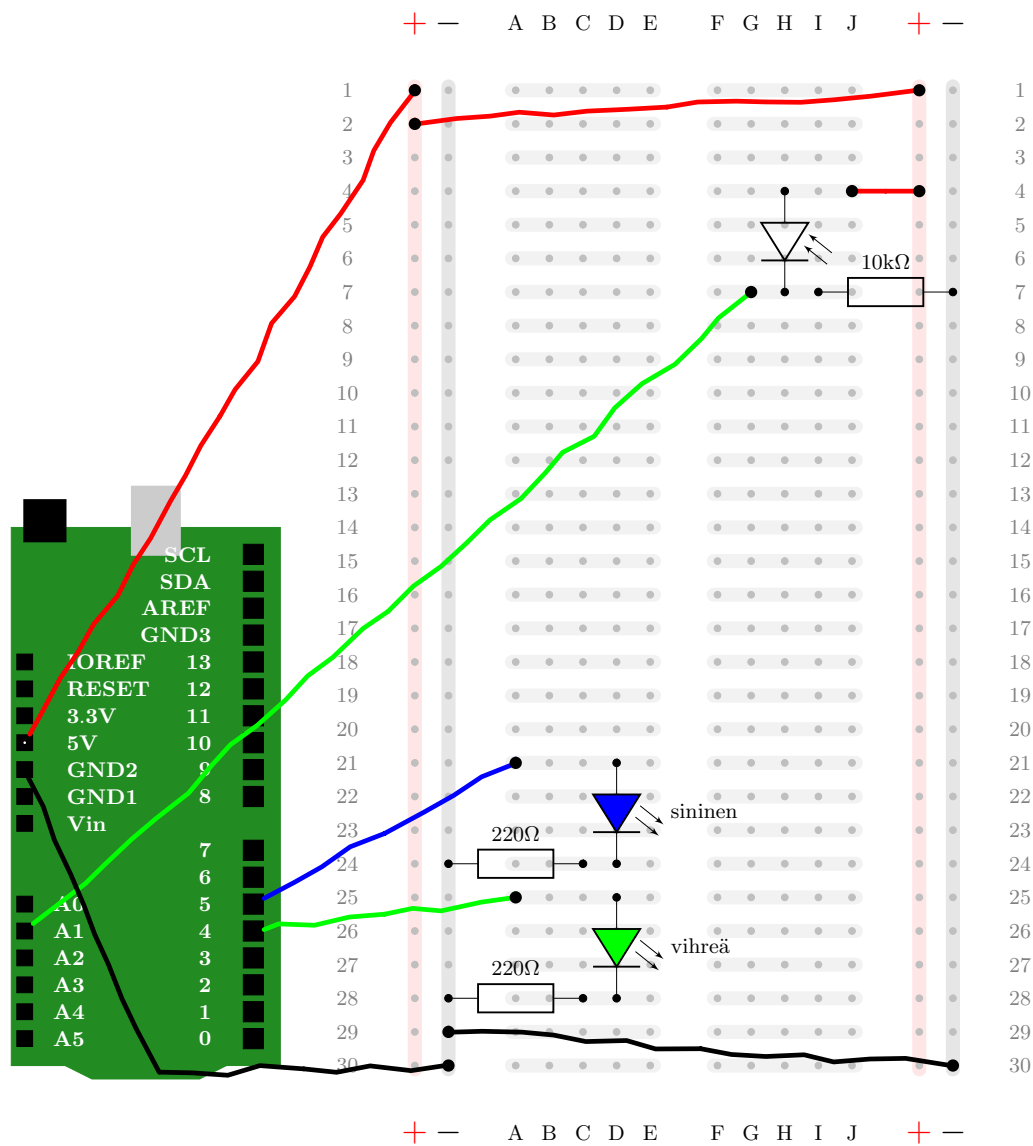
Piirin toiminta

Lisätään havainnointia varten kaksi LEDiä, joilla saadaan tietää onko huoneessa valoisaa vai pimeää.

**HUOM!**

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

Lisätään LEDit sekä vastukset kuvan mukaisesti.



Lisätään taas LEDit koodiin.

```
// Perusvalon aiheuttama lukema
int valo = 400; // Muokkaa arvoksi, jonka sait edellisestä tehtävästä

void setup() {
  Serial.begin(9600);
  pinMode(5, OUTPUT); // Sininen LED
  pinMode(4, OUTPUT); // Vihreä LED
}

void loop() {
  int valon = analogRead(A1); // Luetaan anturin arvo portista A1
  Serial.print("Anturin arvo: "); // Selitys
  Serial.println(valon); // ja tulostetaan se

  // Mita tehdään?
  if (valon < valo){
    digitalWrite(5, HIGH); // Sytytetään sininen valo
    digitalWrite(4, LOW); // Sammutetaan vihreä valo
  } else if (valon >= valo && valon < 2*valo) {
    digitalWrite(5, LOW); // Sammutetaan sininen valo
    digitalWrite(4, LOW); // Sammutetaan vihreä valo
  } else if (valon >= 2*valo) {
    digitalWrite(5, LOW); // Sammutetaan sininen valo
    digitalWrite(4, HIGH); // Sytytetään vihreä valo
  }

  delay(1000); // Odotetaan, että ehditaan lukea
}
```

Tutki!

Peitä kädellä valoanturi. Kumpi valo syttyy?

Lisää valoa esimerkiksi taskulampulla. Kumpi valo syttyy?

Sininen valo syttyy kun on pimeää ja vihreä kun on valoisaa.

3.3 Valo- ja lämpöanturit

Tarvikkeet

- Edellisten kahden kohdan piirit (3.1.2 ja 3.2.2) rakennettuna

Piirin toiminta

Yhdistetään valo- ja lämpöanturit.

HUOM!

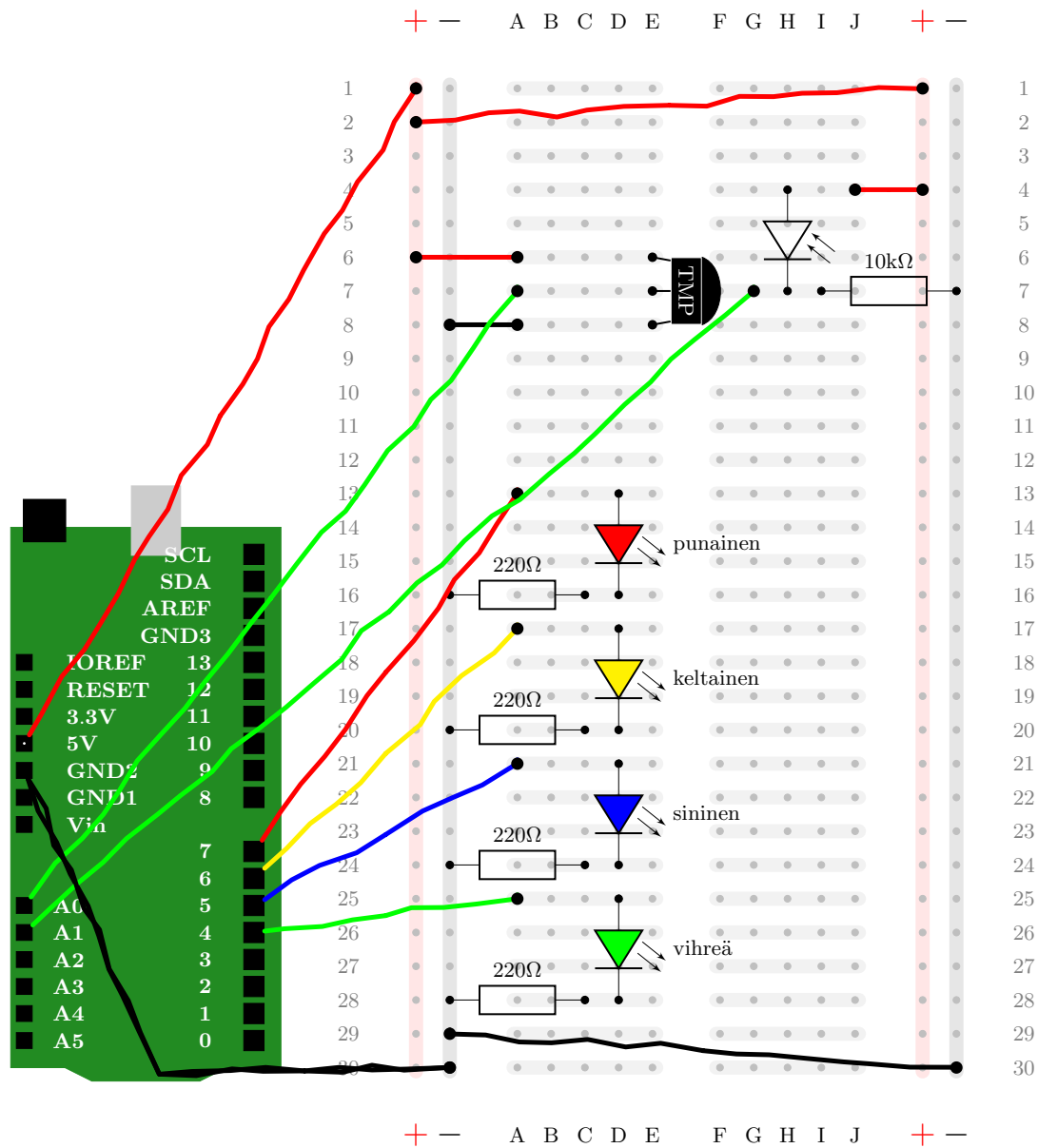
Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

Jos sinulla on edelleen jäljellä kohtien 3.1.2 ja 3.2.2 piirit, sinulla pitäisi olla seuraavan sivun kuvan mukaiset kytkennät. Jos olet kuitenkin purkanut osan piireistä, kokoa se uudelleen kuvan mukaisesti.

Haaste!

Suunnittele oma hälytysjärjestelmä, joka reagoi joko valon tai lämmön tai molempien muutoksiin.

Seuraavana on kuva, jossa on yhdistetty molemmat piirit ja sitä seuraa koodi, joka yhdistää jo tekemämme koodit. Voit käyttää sitä pohjana suunnitelmallesi.



```
// Huoneen normaalit lukemat
int valo = 400; // Muokkaa saamaksesi arvoksi
const float huoneLamp = 20.0; // Muokkaa saamaksesi arvoksi

void setup() {
  Serial.begin(9600);
  pinMode(7, OUTPUT); // Punainen LED
  pinMode(6, OUTPUT); // Keltainen LED
  pinMode(5, OUTPUT); // Sininen LED
  pinMode(4, OUTPUT); // Vihreä LED
}
```

```
}

void loop() {
  int valoa = analogRead(A1); // Luetaan valoanturin arvo portista A1
  Serial.print("Valoanturin arvo: "); // Selitys
  Serial.println(valoa); // ja tulostetaan se

  int sensorVal = analogRead(A0); // Luetaan anturin arvo portista A0
  Serial.print("Lämpötila: "); // lisataan selitys luvulle
  float voltage = (sensorVal/1024.0)*5.0; // Muunna anturin arvo
    volteiksi
  float temp = (voltage-0.5)*100; // Muunnetaan jännite Celcius-
    asteiksi
  Serial.print(temp); // ja tulostetaan lämpötila
  Serial.println(" C."); // Celcius-astetta ja vaihdetaan rivia

  // Mita tehdään?
  if (valoa < valo){
    digitalWrite(5,HIGH); // Sytytetään sininen valo
    digitalWrite(4,LOW); // Sammutetaan vihreä valo
  } else if (valoa >= valo && valoa < 2*valo) {
    digitalWrite(5,LOW); // Sammutetaan sininen valo
    digitalWrite(4,LOW); // Sammutetaan vihreä valo
  } else if (valoa>=2*valo) {
    digitalWrite(5,LOW); // Sammutetaan sininen valo
    digitalWrite(4,HIGH); // Sytytetään vihreä valo
  }

  if (temp < huoneLamp-2){
    // Huoneessa alkaa olla kylmä
    digitalWrite(7,LOW); // Sammutetaan punainen LED tarvittaessa
    digitalWrite(6,HIGH); // Sytytetään keltainen LED
  }else if(temp>=huoneLamp-2 && temp <huoneLamp+2){
    // Normaalihuoneen lamppu
    digitalWrite(7,LOW); // Sytytetään punainen valo
    digitalWrite(6,LOW); // Sytytetään keltainen valo
  }else if (temp>huoneLamp+2) {
    // Lamppu on noussut liikaa, varoitetaan pelkällä punaisella valolla
    digitalWrite(7,HIGH); // Sytytetään punainen valo
    digitalWrite(6,LOW); // Sammutetaan keltainen valo
  }
}
```

```
    delay(1000); // Odotetaan, että ehditaan lukea  
}
```

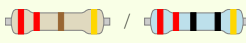
Koodaa!

Tähän tehtävään ei ole malliratkaisua

4. LCD-näyttö

4.1 LCD-näytön kytkentä

Tarvikkeet

- Koekytkentälevy
- Arduino UNO
- Potentiometri
- Hyppylankoja
- Vastus 220Ω: 
- LCD-näyttö

Piirin toiminta

Tutustutaan LCD-näytön käyttämiseen. Kirjoitetaan oma viesti näytölle. Potentiometrin käyttäminen ja sen avulla näytön kirkkauden säätäminen.

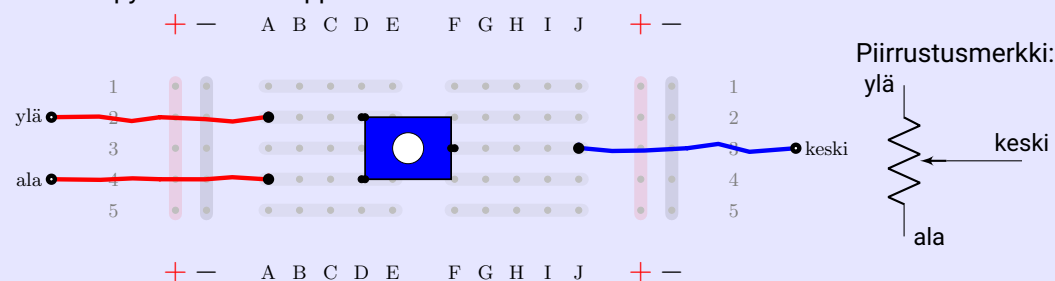
HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

Huomaa, että LCD-näytön kytkemiseen tarvitaan useita hyppylankoja ja se vaatii tarkkuutta!

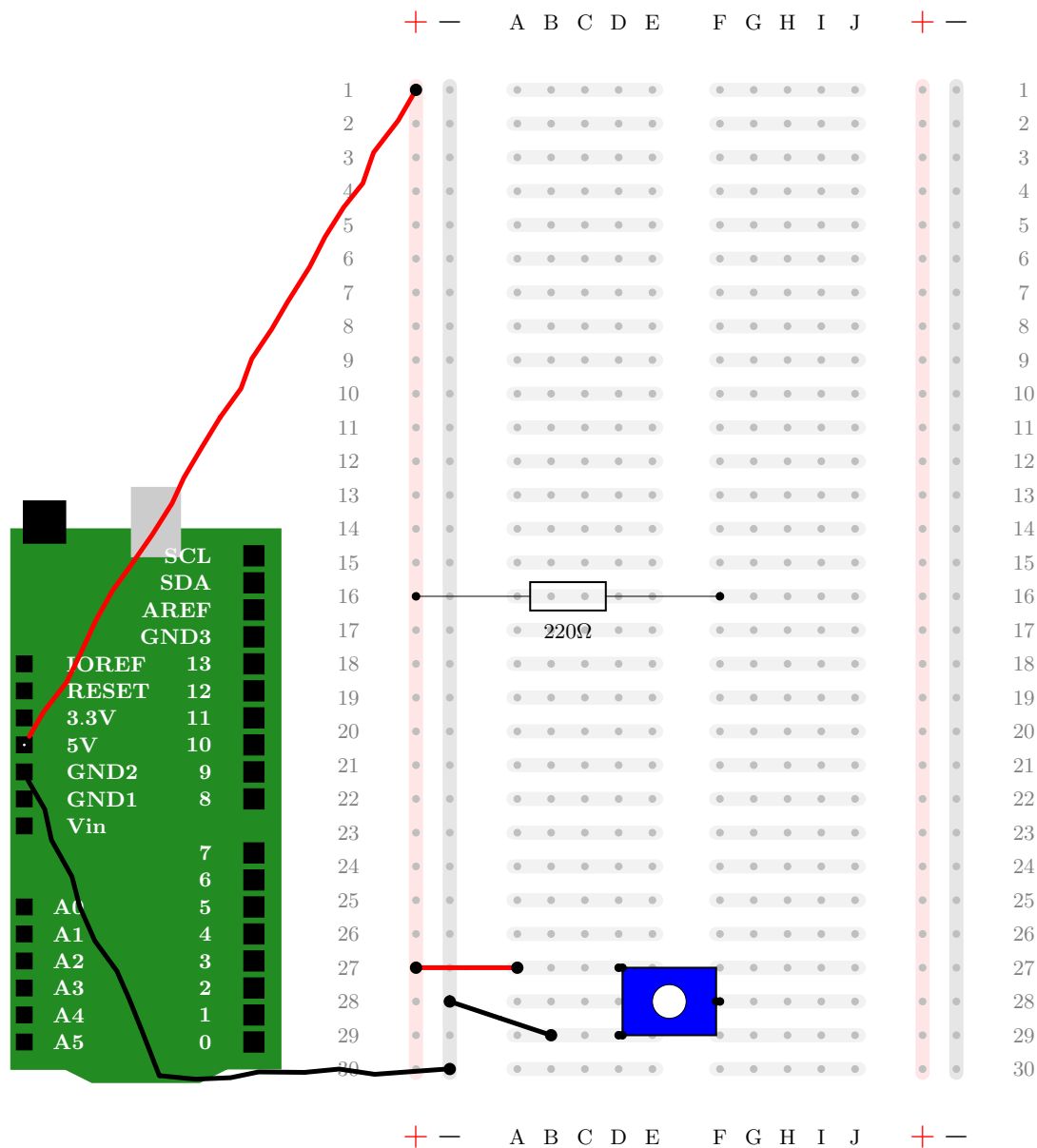
Potentiometrin kytkeminen

Potentiometri on säädettävä vastus, jossa on kolme jalkaa, kaksi toisella puolella ja yksi toisella. Se kytketään uran ylitse, niin että kaksi jaloista ovat toisella puolella ja yksi toisella puolella. Potentiometrin mukana tulee valkoinen säätönuppi, jolla vastuksen arvoa voidaan säädellä pyörittämällä nuppia.

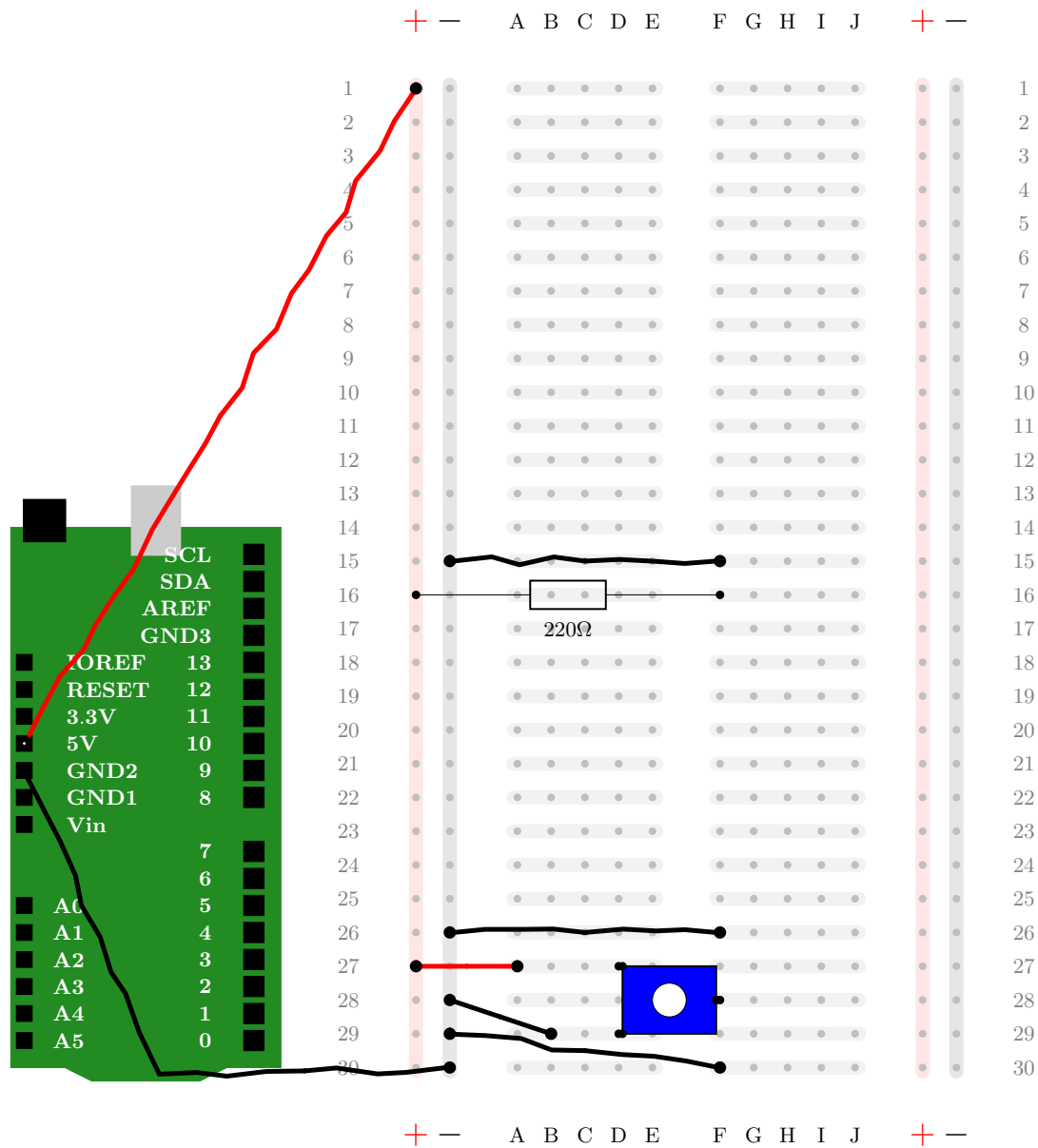


Mitattaessa vastuksen arvoa välillä ylä-ala, vastuksen arvo pysyy vakiona. Jos taas käytetään vastusta välillä ylä-keski tai keski-ala, vastuksen arvoa voidaan muuttaa pyörittämällä säätönuppia.

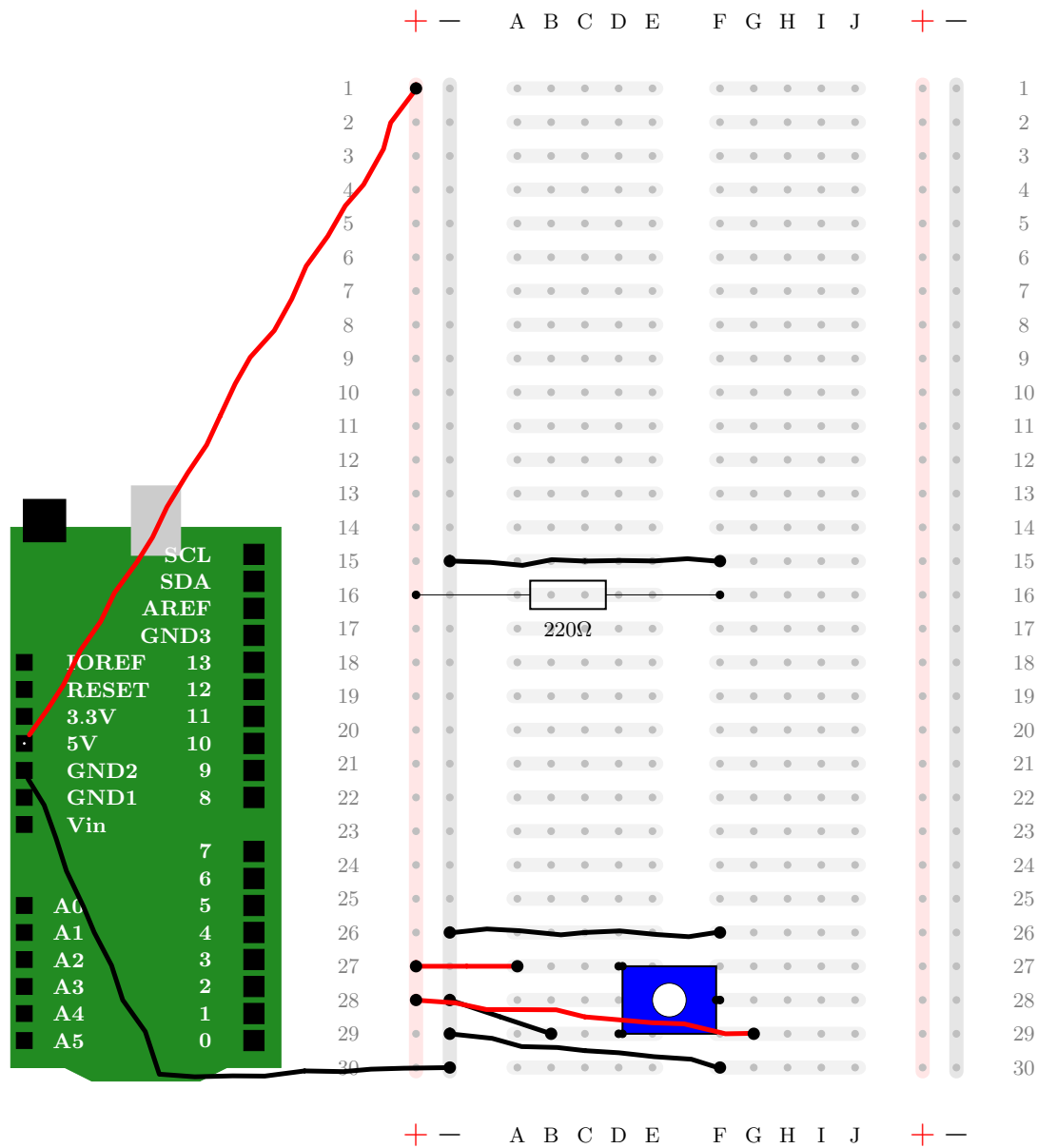
Kytetään näyttöön tarvittavat asiat osissa koekytkentälevylle. Ole tarkkana rivien kanssa!



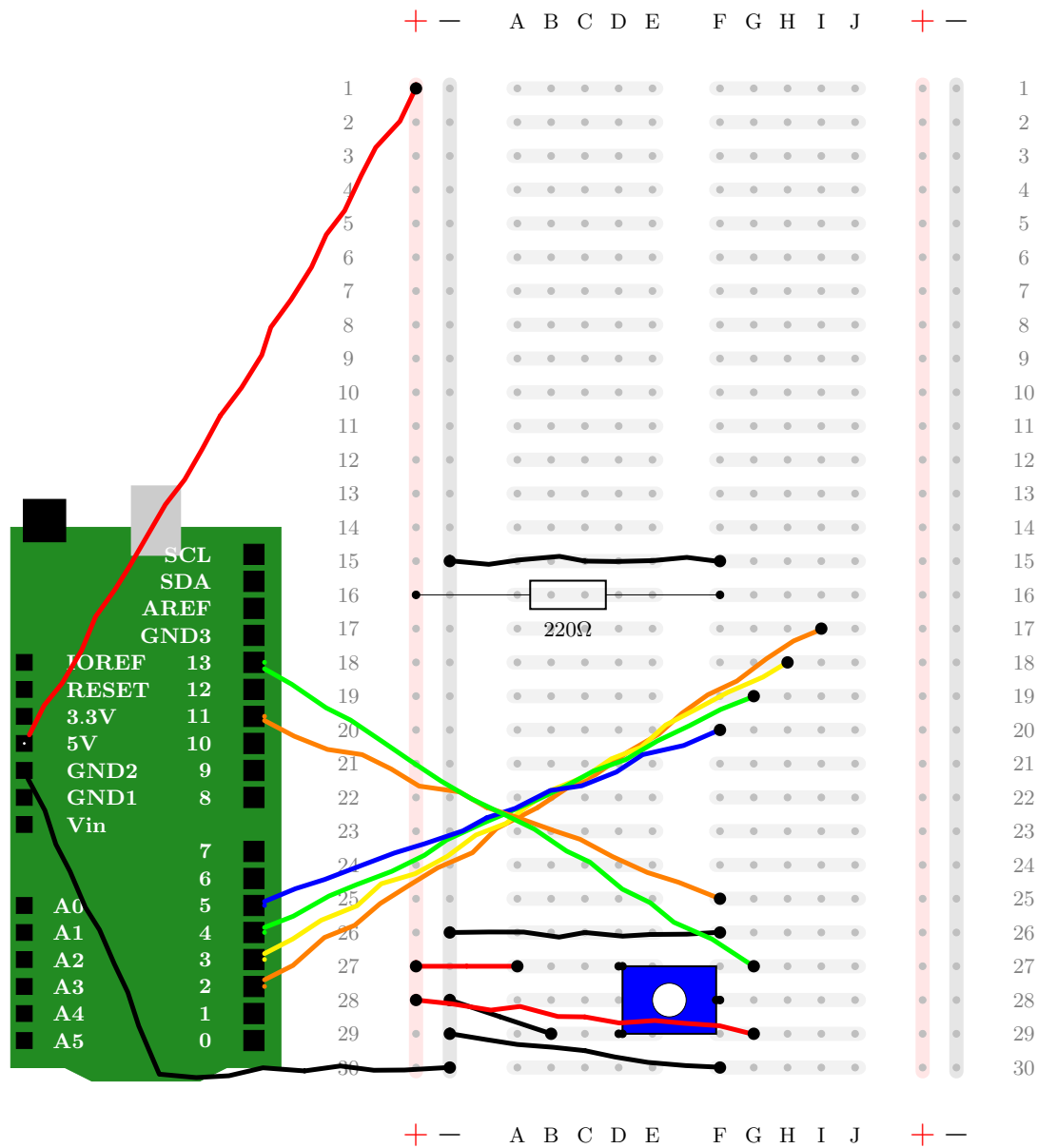
Kytetään ensin potentiometri pisteisiin D27, D29 ja F28. Kytke rivi 27 vasen puoli käyttöjännitteeseen (punainen + rivi) ja rivi 29 vasen puoli maahan. Kytetään sitten vastus käyttöjännitteeseen (punainen +, vasen laita) ja F16 välille. Muista myös kytkeä käyttöjännite vasemmalle + riville ja maa vasemmalle - riville.



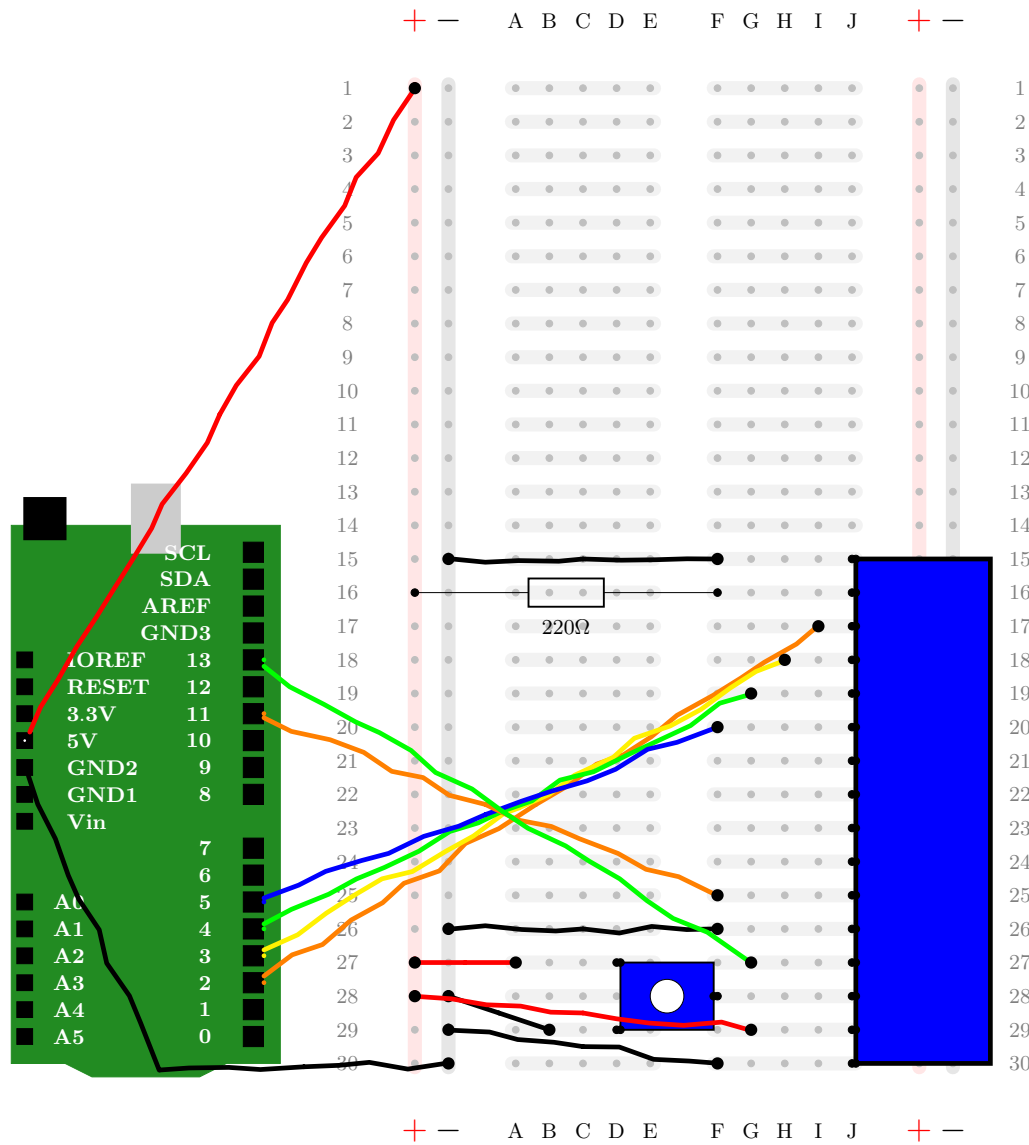
Kytetään sitten maat: rivi 26 oikea puoli maahan, rivi 30 oikea puoli maahan, ja rivi 15 oikea puoli maahan.



Kytetään sitten loput käyttöjännitteet: rivi 29 oikea puoli vasempaan plus-riviin (5V).



Kytetään Arduinon puolelta pinni 2 oikealle riville 17, pinni 3 oikealle riville 18, pinni 4 oikealle riville 19 ja pinni 5 oikealle riville 20. Lisäksi pinni 11 oikealle riville 25 ja pinni 13 oikealle riville 27.



Viimeiseksi lisään LCD-näyttö kiinni pinneihin J15-J30.

Seuraavaksi koodataan Arduinolla näytölle tekstiä kahteen riviin. LCD-näytöllä on kaksi riviä, joiden pituus on 16 merkkiä riviä kohti.

Vinkkejä Arduinolla koodaamiseen!

```
1 // LCD-nayton kayttaminen onnistuu kirjaston avulla:
2 #include <LiquidCrystal.h> // Lisataan kirjasto
3 // Arduinolle pitää kertoa missä pinneissa naytto on kiinni:
4 LiquidCrystal lcd(13,11,5,4,3,2); // Digital Pinnit joissa kiinni
5
6 // Taman jälkeen naytolle voidaan tulostaa:
7 lcd.print("Viesti.");
8 // Naytto voidaan tyhjentää
9 lcd.clear();
10 // Osoittimen paikkaa voidaan siirtää:
11 lcd.setCursor(luku1, luku2);
12 // luku1 kertoo kuinka mones nayton osa (0-15)
13 // luku2 kertoo kumpi rivi, 0 = ylempi, 1 = alempi
```

Kopioi koodi ja testaa, että saat näytölle näkyviin tekstin "Tervetuloa (Nimesi)". Jos näytöllä ei näy mitään, niin kokeile pyörittää potentiometriä nupin avulla. Potentiometri säätää näytön kirkkautta.

```
#include <LiquidCrystal.h> // Lisataan kirjasto
LiquidCrystal lcd(13,11,5,4,3,2); // Digital Pinnit joissa kiinni

void setup() {
  lcd.begin(16,2); // Kaynnistetaan naytto
  lcd.print("Tervetuloa"); // Kirjoitetaan ekalle riville
}

void loop() {
  lcd.setCursor(0,1); // Siirrytaan toiselle riville
  lcd.print("Nimi"); // Kirjoita oma nimesi tahan
}
```

Seuraavaksi testataan, miten saadaan aikaan liikkuva teksti toiselle riville.

```
#include <LiquidCrystal.h> // Lisataan kirjasto
LiquidCrystal lcd(13,11,5,4,3,2); // Digital Pinnit joissa kiinni

char * viesti = "Tanaan on ohjelmointipaiva.";

void setup() {
  lcd.begin(16,2);
  lcd.print("Tervetuloa");
}

void loop() {
  // Naytolle mahtuu 16 merkkiä per rivi
  for (int letter = 0; letter <= strlen(viesti); letter++) {
    naytaViesti(0,letter);
  }
}

void naytaViesti(int printStart, int startLetter) {
  lcd.setCursor(printStart,1);
  for (int letter = startLetter; letter<=startLetter+15; letter++) {
    if (letter <=strlen(viesti)-1){
      lcd.print(viesti[letter]);
    } else {
      lcd.print(" ");
    }
  }
  lcd.print(" ");
  delay(300); // Lisataan viive, etta ehdimme lukea
}
```

Nyt alemmalla rivillä pitäisi liikkua määrittelemänne viesti! Voit muokata viestin liikkumisnopeutta muokkaamalla viivettä. Muokkaa koodia oman tervehdysten lähettämiseen!

Koodaa!

```
1 #include <LiquidCrystal.h> // Lisataan kirjasto
2 LiquidCrystal lcd(13,11,5,4,3,2); // Digital Pinnit joissa kiinni
3
4 // Tata muokkaamalla saadaan eri liikkuvia viesteja.
5 char * viesti = "Tanaan on ohjelmointipaiva.";
```

```
6
7 void setup() {
8   lcd.begin(16,2);
9   lcd.print("Tervetuloa"); // Tata muokkaamalla ylarivi pysyy
    samana
10 }
11
12 void loop() {
13   // Naytolle mahtuu 16 merkkia per rivi
14   for (int letter = 0; letter <= strlen(viesti); letter++) {
15     naytaViesti(0,letter);
16   }
17 }
18
19 void naytaViesti(int printStart, int startLetter) {
20   lcd.setCursor(printStart,1);
21   for (int letter = startLetter; letter<=startLetter+15; letter++)
22     {
23       if (letter <=strlen(viesti)-1){
24         lcd.print(viesti[letter]);
25       } else {
26         lcd.print(" ");
27       }
28     }
29   lcd.print(" ");
30   delay(300); // Lisataan viive, etta ehdimme lukea
```

4.2 Lämpötilan mittaaminen ja näyttäminen näytöllä

Tarvikkeet

- Edellisen kohdan piiri rakennettuna
- Lämpötila-anturi (TMP36)
- Hyppylankoja

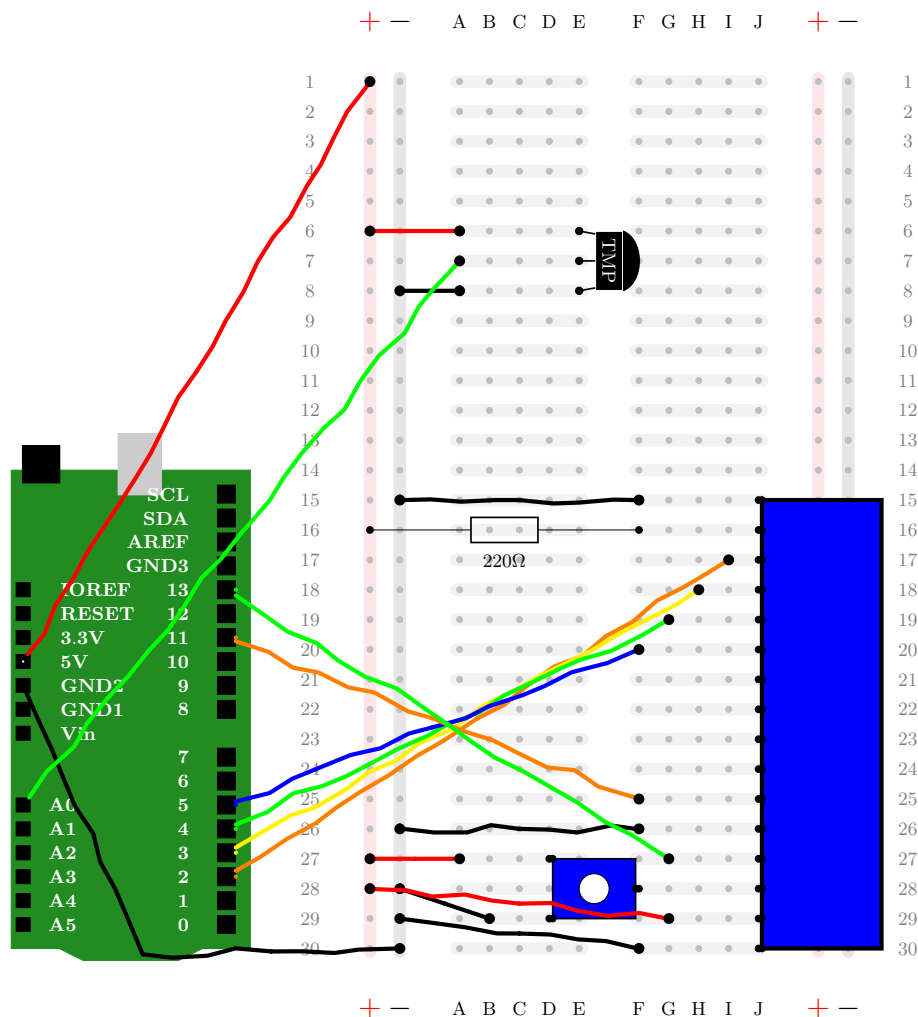
Piirin toiminta

Mitataan huoneen lämpötila ja tulostetaan se LCD-näytölle.

HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

Lisätään lämpötila-anturi piiriin kuvan mukaisesti ja liitetään sen mittaus porttiin A0.



Kuten jo aiemmin, kun mittasimme lämpötilaa, voimme tehdä samoin nyt ja tulostaa lämpötilan serial portin sijaan näytölle.

```
#include <LiquidCrystal.h> // Lisataan kirjasto
LiquidCrystal lcd(13,11,5,4,3,2); // Digital Pinnit joissa kiinni

void setup() {
  lcd.begin(16,2); // Aloitetaan kaytto
  lcd.print("Lampotila"); // Tulostetaan ensimmäiselle riville
    lampotila
}

void loop() {
  // Mitataan lampotila ja muunnetaan se asteiksi
  int sensorVal = analogRead(A0);
  float voltage = (sensorVal/1024.0)*5.0;
  float temp = (voltage-0.5)*100;
  // Ja tulostetaan se
  lcd.setCursor(0,1);
  lcd.print(temp);
  // Odotetaan arvojen päivittämisen valilla, etta luvut eivät hypi
  delay(1000);
}
```

4.3 LCD-näyttö ja vaihtuvat viestit

Tarvikkeet

- Edellisen kohdan piiri rakennettuna
- Kaltevuusanturi
- Vastus 10k Ω :  / 

Piirin toiminta

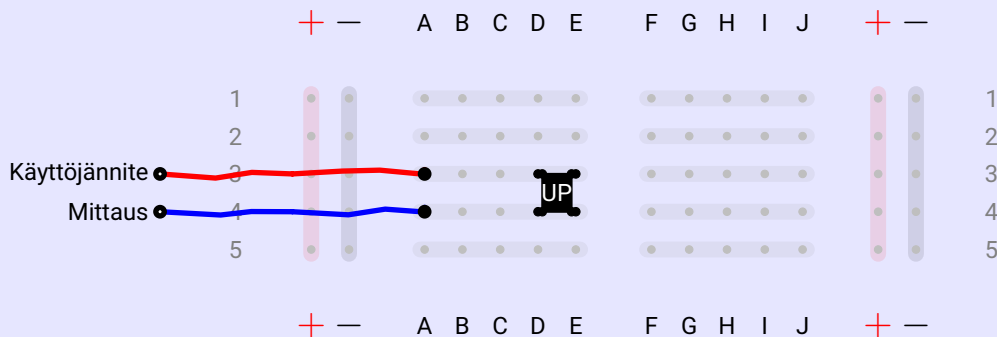
Vaihdetaan LCD-näytön tekstiä kun havaitaan, että piiriä on kallistettu.

HUOM!

Aina kun rakennat tai muutat piiriä, pidä Arduino irrotettuna tietokoneesta!

Kaltevuusanturin kytkeminen

Kaltevuusanturissa on neljä jalkaa, ja se vaatii kaksi riviä koekytkentälevystä. Kaltevuusanturi näyttää suorakulmaiselta laatikolta, jonka lähes neliönmuotoisella päällä lukee "UP" ja on nuoli ylös. Tämä rivi, jolla nuoli on, kytketään käyttöjännitteeseen ja toinen rivi kytketään vastuksen kautta maahan. Mittaustulos voidaan lukea samalta riviltä mille vastus on kytketty.



Vinkkejä Arduinolla koodaamiseen!

```

1 // Satunnaisen luvun arpominen
2 int satunnainen = random(luku); // satunnainen kokonaisluku
   valilta 0-annettu luku
3 // Ehtolause: ei yhtä suuri kuin
4 luku1 != luku2
5 // Vaihtoehtoisia tapauksia
6 switch(luku) {
7     case 0:
8         // mita tehdään jos luku on nolla
9         break; // lopetetaan

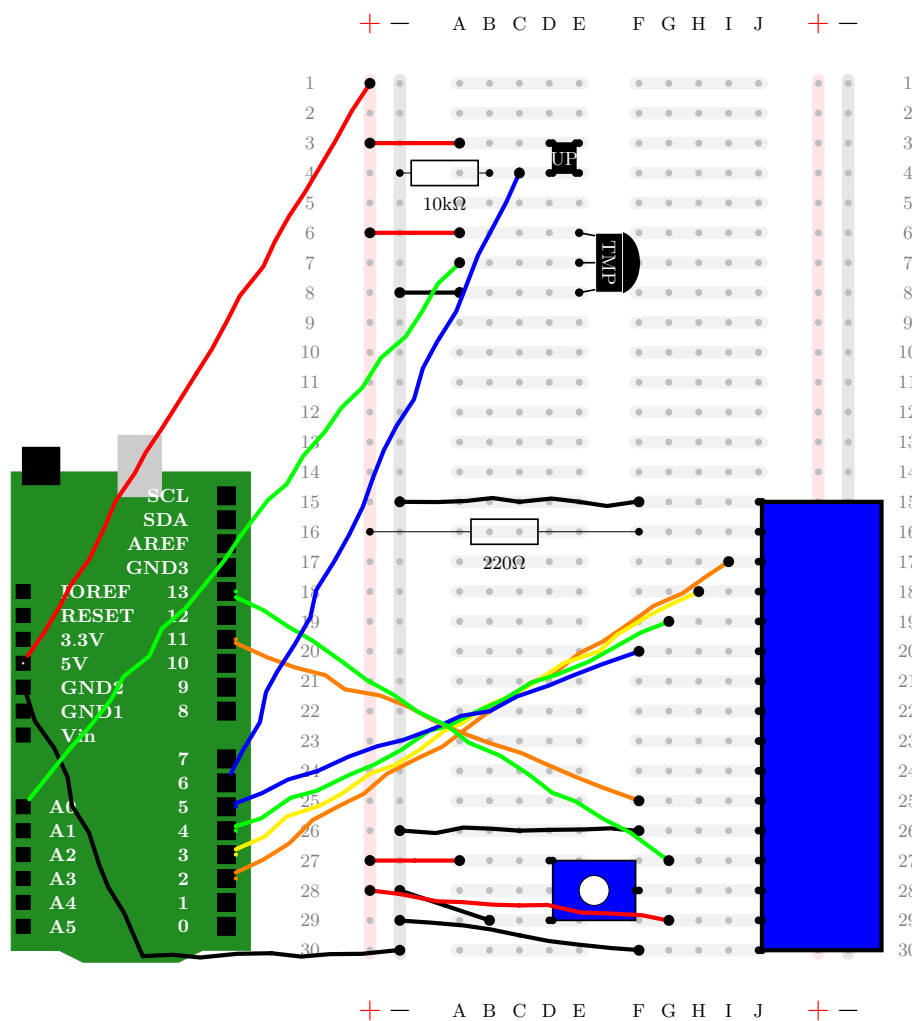
```

```

10 case 1:
11 // kun luku on 1
12 break;
13 // Lisaa case numero: riveja niin monta yhteensa kuin lukusi
   on ja kerro mita tehdään, ja poistu sen jalkeen ulompaan
   koodiin break:n avulla
14 }

```

Kytetään kaltevuusanturi ja vastus piiriin ja mitataan anturin antamaa arvoa pinnin 6 avulla.



Lisätään koodiin kaltevuusanturin lukeminen ja koodataan muutama eri viesti näytöllä näytettäväksi.

```
#include <LiquidCrystal.h> // Lisataan kirjasto
LiquidCrystal lcd(13,11,5,4,3,2); // Digital Pinnit joissa kiinni

int switchState = 0;
int prevSwitchState = 0;
int reply;

void setup() {
  lcd.begin(16,2);
  pinMode(6, INPUT); // Kaltevuusanturi pinnissa 6
  lcd.print("Hetken miete");
}

void loop() {
  // Luetaan kaltevuusanturin tila
  switchState = digitalRead(6);
  // Jos tila ei ole sama kuin edellinen tila, tehdään asioita
  if (switchState != prevSwitchState) {
    if (switchState == LOW) {
      // Arvotaan mita näytetään
      reply = random(4);
      // Tyhjennetään näyttö
      lcd.clear();
      // Asetetaan kursori ensimmäisen rivin alkuun
      lcd.setCursor(0,0);
      // Annetaan eri tapaukset ja mita tehdään milloinkin
      switch(reply){
        case 0:
          lcd.print("Kytke ne");
          lcd.setCursor(0,1);
          lcd.print("sarjaan!");
          break;
        case 1:
          lcd.print("Kytke ne");
          lcd.setCursor(0,1);
          lcd.print("rinnan!");
          break;
        case 2:
          lcd.print("LEDit");
```

```
        lcd.setCursor(0,1);  
        lcd.print("palaa!");  
        break;  
        case 3:  
        lcd.print("Lampotila");  
        lcd.setCursor(0,1);  
        int sensorVal = analogRead(A0);  
        float volt = (sensorVal/1024.0)*5.0;  
        float temp = (volt-0.5)*100;  
        lcd.print(temp);  
        break;  
    }  
}  
}  
// Viimekseksi tallennetaan edellisiksi tilaksi nykyinen tila  
prevSwitchState = switchState;  
}
```

Testaa toiminta nostamalla sitä. Heiluta varovasti koekytkentälevyä, niin että viestit vaihtuvat näytöllä.

Haaste!

Muuta koodia niin, että saat aikaa mystisen kasipallon! Eli, koodaa 8 eri vastausta kysymyksiin, jotka saat esiin ravistamalla kevyesti koekytkentälevyä.

Koodaa!

```
1 #include <LiquidCrystal.h> // Lisataan kirjasto  
2 LiquidCrystal lcd(13,11,5,4,3,2); // Digital Pinnit joissa kiinni  
3  
4 int switchState = 0;  
5 int prevSwitchState = 0;  
6 int reply;  
7  
8 void setup() {  
9     lcd.begin(16,2);  
10    pinMode(6, INPUT); // Kaltevuusanturi pinnissa 6  
11    lcd.print("Mystinen");  
12    lcd.setCursor(0,1);  
13    lcd.print("Kasipallo!");  
14 }
```

```
15
16 void loop() {
17     // Luetaan kaltevuusanturin tila
18     switchState = digitalRead(6);
19     // Jos tila ei ole sama kuin edellinen tila, tehdään asioita
20     if (switchState != prevSwitchState) {
21         if (switchState == LOW) {
22             // Arvotaan mita näytetään
23             reply = random(8);
24             switch(reply) {
25                 case 0:
26                     lcd.clear();
27                     lcd.setCursor(0,0);
28                     lcd.print("Tietokone sanoo");
29                     lcd.setCursor(0,1);
30                     lcd.print("ei.");
31                     break;
32                 case 1:
33                     lcd.clear();
34                     lcd.setCursor(0,0);
35                     lcd.print("Tottakai!");
36                     break;
37                 case 2:
38                     lcd.clear();
39                     lcd.setCursor(0,0);
40                     lcd.print("Ei missään");
41                     lcd.setCursor(0,1);
42                     lcd.print("nimessa");
43                     break;
44                 case 3:
45                     lcd.clear();
46                     lcd.setCursor(0,0);
47                     lcd.print("Kuuma!");
48                     break;
49                 case 4:
50                     lcd.clear();
51                     lcd.setCursor(0,0);
52                     lcd.print("Kylmä!");
53                     break;
```

```
54     case 5:
55         lcd.clear();
56         lcd.setCursor(0,0);
57         lcd.print("Mahdollisesti!");
58         break;
59     case 6:
60         lcd.clear();
61         lcd.setCursor(0,1);
62         lcd.print("Huomenna!");
63     case 7:
64         lcd.clear();
65         lcd.setCursor(0,0);
66         lcd.print("Lampotila");
67         lcd.setCursor(0,1);
68         int sensorVal = analogRead(A0);
69         float volt = (sensorVal/1024.0)*5.0;
70         float temp = (volt-0.5)*100;
71         lcd.print(temp);
72         break;
73     }
74 }
75 }
76 // Viimekseksi tallennetaan edelliseksi tilaksi nykyinen tila
77 prevSwitchState = switchState;
78 }
```

Keskustelua voi nostaa esimerkiksi siitä, kuinka satunnainen random() oikeastaan on. Kannattaa muistaa myös, että Arduinossa on piirissä reset-nappula tietokone-liitännän vieressä, jos haluat palauttaa koodin takaisin alkupisteeseen.

A. Komponenttipankki

Tässä osuudessa on kerätty yhteen tekstistä löytyvät komponenttilaatikot.

Vastuksen arvo

Jos sinulla ei ole käytössä yleismittaria oikean vastuksen arvon mittaamiseen, tai haluat harjoitella vastusten värikoodien lukemista, vastusten värikoodit löydät sivulta 79.

Nyt tarvitaan 220Ω vastus eli jompi kumpi alla olevista vaihtoehdoista:

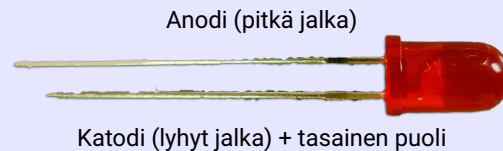
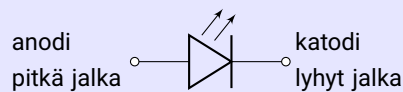


LEDin kytkeminen

Huomaa, että on merkitystä kummin päin LED kytetään koekytkentälevylle!

LEDissä on kaksi jalkaa, joista toinen on pidempi. LEDin muovi on toiselta puolelta kaareva, ja toiselta siinä on suora leikkaus, lyhyempi jalka on suoran leikkauksen puolella. Lyhyemmän jalan nimi on katodi ja pidemmän jalan anodi.

Piirrosmerkissä:

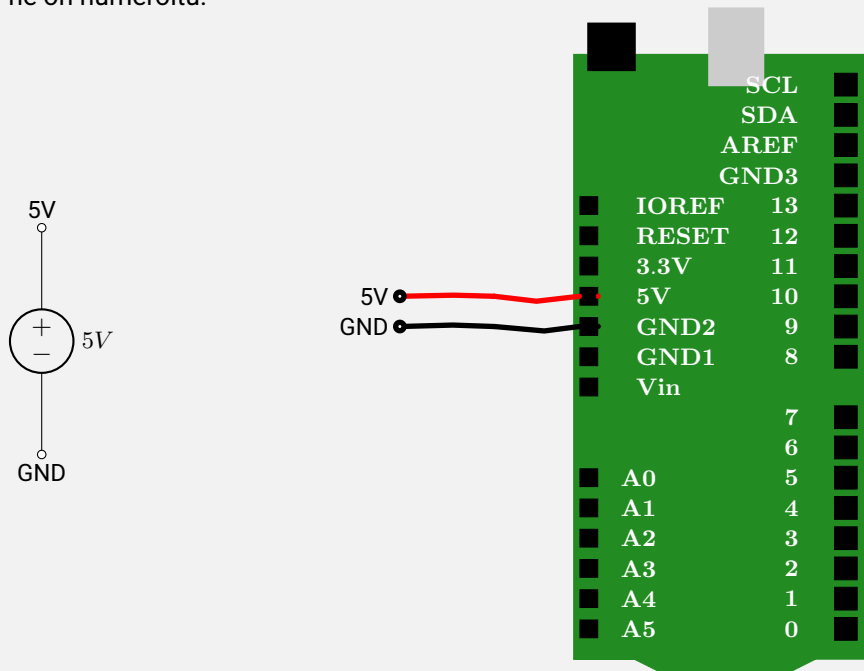


LED, eli valoa säteilevä diodi päästää virtaa läpi vain toiseen suuntaan kuten diodit. Jos anodille kytketään tarpeeksi suuri jännite verrattuna katodiin, virta kulkee ja valo palaa. Jos taas katodilla on suurempi jännite kuin anodilla, virtaa ei kulje ja LED ei pala. Jos anodilla ei ole tarpeeksi suuri jännite verrattuna katodiin, virta ei kulje. Riippuen LEDistä, vaadittu kynnysjännite voi vaihdella ja on tyypillisesti 1,5-4,5 voltia.

Jännitelähteen kytkeminen

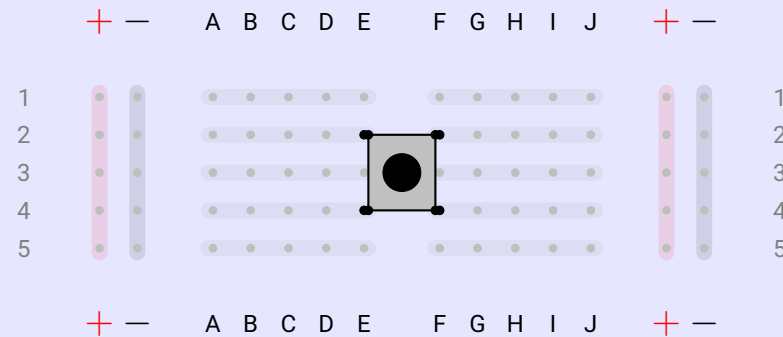
Jännitelähde on Arduinossa sisäänrakennettuna. Jotta voimme kytkeä jännitelähteen piiriin, tarvitsemme jännitelähteen molemmat päät: +5V ja maan (GND), kytkemällä vain toinen piiriin emme saa aikaan haluttua vaikutusta.

Arduino levyllä on kolme eri GND-riviä, ja mikä tahansa niistä käy. Kuvassa selvyden vuoksi ne on numeroitu.



Painonapin kytkeminen

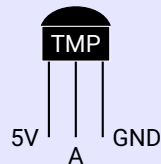
Arduinon mukana tulevassa painonapissa on neljä pinniä, ja asetetaan keskellä olevan uran ylitse. Jos nappia ei paineta, niin A-E ja F-J rivillä 2 ovat samaa pistettä, mutta eri pistettä kuin rivin 4 (A-E) ja (F-J). Jos nappia painetaan, niin molempien rivien (2 ja 4) kaikki pisteet ovat kiinni toisissaan.



Lämpötila-anturi

Huomaa, että on merkitystä miten päin lämpötila-anturi kytketään piiriin!

Jos tasainen sivu on kohti sinua, silloin vasemman puoleisin jalka on käyttöjännite (5V), keskimmaiselta jalalta (A niin kuin anturi) voidaan lukea tulos ja oikean puoleisin jalka kytketään maahan (GND).



Jos kytket anturin väärinpäin, se voi kuumentua. Ole siis tarkkana!

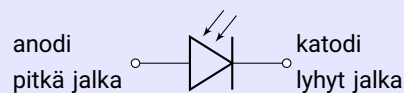
Valoanturin kytkeminen

Huomaa, että on merkitystä kummin päin valoanturi kytketään koekytkentälevylle!

Valoanturissa on kaksi jalkaa, joista toinen on pidempi. Valodiodi näyttää kirkkaalta LEDiltä, mutta siitä puuttuu pyöristetty yläosa, eli yläpinta on tasainen (toisin kuin LEDeillä, joilla se on pyöreä). Piirrosmerkissä nuolet tulevat kohti diodia, eli olemme ottamassa valoa sisään (emme lähettämässä sitä ulos kuten LEDin kanssa).

Valoanturin muovi on toiselta puolelta kaareva, ja toiselta siinä on suora leikkaus, lyhyempi jalka on suoran leikkauksen puolella. Lyhyemmän jalan nimi on katodi ja pidemmän jalan anodi.

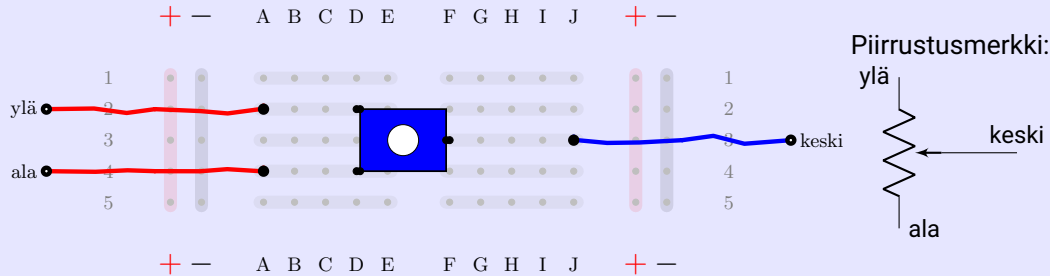
Piirrosmerkissä:



Valodiodi kytketään myös aina vastuksen kanssa sarjaan, nyt käytössä on 10k Ω vastus.

Potentiometrin kytkeminen

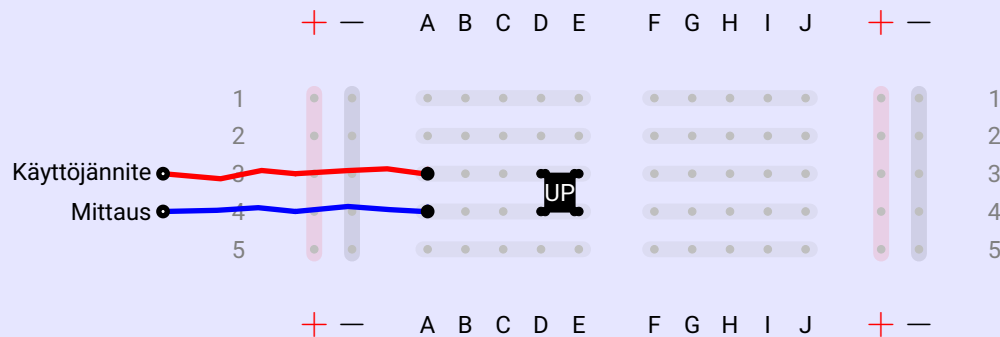
Potentiometri on säädettävä vastus, jossa on kolme jalkaa, kaksi toisella puolella ja yksi toisella. Se kytketään uran ylitse, niin että kaksi jaloista ovat toisella puolella ja yksi toisella puolella. Potentiometrin mukana tulee valkoinen säätönuppi, jolla vastuksen arvoa voidaan säädellä pyörittämällä nuppia.



Mitattaessa vastuksen arvoa välillä ylä-ala, vastuksen arvo pysyy vakiona. Jos taas käytetään vastusta välillä ylä-keski tai keski-ala, vastuksen arvoa voidaan muuttaa pyörittämällä säätönuppia.

Kaltevuusanturin kytkeminen

Kaltevuusanturissa on neljä jalkaa, ja se vaatii kaksi riviä koekytkentälevystä. Kaltevuusanturi näyttää suorakulmaiselta laatikolta, jonka lähes neliönmuotoisella päällä lukee "UP" ja on nuoli ylös. Tämä rivi, jolla nuoli on, kytketään käyttöjänniteeseen ja toinen rivi kytketään vastuksen kautta maahan. Mittaustulos voidaan lukea samalta riviltä mille vastus on kytketty.



LCD-näyttö

Tämä on tarkemmin esitelty kappaleessa 4.1 ja tarkemmin kuvassa sivulla 58. Tällöin LCD-näyttöä ohjataan seuraavilla komennoilla.

```
#include <LiquidCrystal.h> // Lisataan kirjasto
LiquidCrystal lcd(13,11,5,4,3,2); // Digital Pinnit joissa kiinni

void setup() {
  lcd.begin(16,2); // Kaynnistetaan naytto
  lcd.print("Tervetuloa"); // Kirjoitetaan ekalle riville
}

void loop() {
  lcd.setCursor(0,1); // Siirrytaan toiselle riville
  lcd.print("Nimi"); // Kirjoita oma nimesi tahan
}
```

B. Vinkit Arduinolla koodaamiseen

Tässä osuudessa on koottu yhteen kaikki vinkit paketin osalta.

Vinkkejä Arduinolla koodaamiseen!

Seuraavassa esitellään Arduinolla koodaamiseen liittyviä vinkkejä.

```
1 // Kaksi kauttaviivaa aloittaa kommentin. Tata ei siis lueta
   koodissa vaan on tiedoksi koodin lukijalle
2
3 pinMode(luku, tyyppi) // Talla voit saataa pinnien 0-13 tekoa
   laittamalla sanan luku tilalle luvun jolta rivilta haluat
   komentaa
4 // Tyyppin tilalle voit kirjoittaa joko OUTPUT eli Arduino lähettää
   signaalia
5 // tai INPUT jolloin Arduino vastaanottaa signaalin
6
7 digitalWrite(luku, arvo) // Talla voit muuttaa pinniin luku
   kirjoitettua arvoa
8 // arvo HIGH vastaa 5V jännitettä
9 // arvo LOW vastaa 0V jännitettä
10
11 delay(luku) // Odottaa luvun verran millisekunteja
```

Vinkkejä Arduinolla koodaamiseen!

```
1 muuttuja = digitalRead(luku); // Lukee pinnin luku arvon ja
   tallentaa sen muuttujaan muuttuja
2 // Huomaa etta muuttujalla pitaa olla annettuna tyyppi
3 // Esimerkiksi bool:
4 bool muuttuja = digitalRead(luku);
5 // bool voi olla joko HIGH tai LOW sen mukaan mita tassa
   tapauksessa luettiin piirista
6 // bool on kateva tilanteissa joissa arvo on joko HIGH tai LOW
7
8 // Ehtorakenne
9 // Tarkastetaan muuttujan arvo.
10 if (muuttuja == HIGH) {
11     // jos muuttujan arvo on HIGH tehdään koodi tassa valissa
```

```
12 } else {  
13     // muuten (eli kun LOW) tehdään mita tassa kirjoitetaan  
14 }
```

Vinkkejä Arduinolla koodaamiseen!

```
1 bool ledStatus = true; // Luodaan muuttuja, jonka arvo on totta  
2  
3 ledStatus = !ledStatus; // Muuttaa muuttujan ledStatus arvoksi  
    arvon epatosi (jos arvo oli tosi) tai tosi (jos arvo oli  
    epatosi)
```

Vinkkejä Arduinolla koodaamiseen!

```
1 // Seuraava rivi lisataan setup() funktioon  
2 attachInterrupt(digitalPinToInterrupt(luku), keskeytys, FALLING);  
3 // attachInterrupt on komento jolla luodaan keskeytys  
4 // digitalPinToInterrupt(luku) kertoo mistä pinnista saadaan  
    painallus  
5 // keskeytys on aliohjelmanne nimi, johon siirrytaan  
6 // FALLING on parametri joka tarkoittaa kun nappia painetaan alas  
7 // Vaihtoehtona ovat myös  
8 // LOW kun pinnin luku jännite on 0V  
9 // CHANGE kun pinnin luku jännite muuttuu  
10 // RISING kun pinnin tila muuttuu nollasta viiteen volttiin  
11 // FALLING kun pinnin tila muuttuu viidestä voltista nollaan  
12  
13 // Maaritetaan uusi funktio olemassa olevien peraan  
14 void keskeytys() {  
15     // Mita funktio tekee?  
16     // Esimerkiksi vaihtaa muuttujan ledState arvo:  
17     ledState = !ledState;  
18 }  
19 // Talloin ohjelman alussa pitää olla maaritys  
20 volatile bool ledState = false;  
21 // volatile tarkoittaa, että muuttujan arvo tarkastetaan aina sita  
    kaytettaessa
```

```
22 // muuttujan ledState arvo voi siis muuttua milloin vain ohjelman  
    aikana
```

Vinkkejä Arduinolla koodaamiseen!

```
1 // Seuraava rivi kuuluu setup()-funktion sisälle  
2 Serial.begin(9600); // Aloittaa kommunikoinnin tietokoneen kanssa  
3 // 9600 kertoo kuinka monta bittia sekunnissa lähetetään.  
4  
5 // Kommentoja, joilla voidaan tulostaa  
6 Serial.print(muuttuja); // Tulostetaan muuttujan muuttuja arvo  
7 Serial.print("Teksti:"); // Tulostetaan Teksti:  
8 Serial.println(muuttuja); // Sama kuin muuttujan tulostus, mutta  
    rivinvaihto lopussa  
9 Serial.print("\n"); // Rivinvaihto manuaalisesti
```

Vinkkejä Arduinolla koodaamiseen!

```
1 // Halutaan tehdä monta asiaa riippuen monesta asiasta  
2 if (ehto) {  
3     // Mita tehdään jos ehto pätee  
4 } else if (ehto2) {  
5     // Mita tehdään jos ehto2 pätee  
6 } else if (ehto3) {  
7     // Naita voit lisätä niin monta kuin tarvitset  
8 } else {  
9     // Ja halutessa voit määritellä myös mitä tehdään muulloin  
10 }  
11  
12 // Erilaisia ehtoja  
13 muuttuja < luku // Muuttujan muuttuja arvo on pienempi kuin  
    annettu luku  
14 muuttuja <= luku // Muuttujan muuttuja arvo on pienempi tai yhtä  
    suuri kuin luku  
15 muuttuja > luku // Muuttuja on suurempi kuin luku  
16 // Kaksi ehtoa yhtä aikaa  
17 muuttuja >= luku1 && muuttuja < luku2 // Muuttuja on suurempi tai  
    yhtäsuuri kuin luku1 JA pienempi kuin luku2
```

Vinkkejä Arduinolla koodaamiseen!

```
1 // Satunnaisen luvun arpominen
2 int satunnainen = random(luku); // satunnainen kokonaisluku
   valilta 0-annettu luku
3 // Ehtolause: ei yhtä suuri kuin
4 luku1 != luku2
5 // Vaihtoehtoisia tapauksia
6 switch(luku) {
7     case 0:
8         // mita tehdään jos luku on nolla
9         break; // lopetetaan
10    case 1:
11        // kun luku on 1
12        break;
13    // Lisää case numero: rivejä niin monta yhteensä kuin lukusi
        on ja kerro mita tehdään, ja poistu sen jälkeen ulompaan
        koodiin break:n avulla
14 }
```

Vinkkejä Arduinolla koodaamiseen!

```
1 // Satunnaisen luvun arpominen
2 int satunnainen = random(luku); // satunnainen kokonaisluku
   valilta 0-annettu luku
3 // Ehtolause: ei yhtä suuri kuin
4 luku1 != luku2
5 // Vaihtoehtoisia tapauksia
6 switch(luku) {
7     case 0:
8         // mita tehdään jos luku on nolla
9         break; // lopetetaan
10    case 1:
11        // kun luku on 1
12        break;
13    // Lisää case numero: rivejä niin monta yhteensä kuin lukusi
        on ja kerro mita tehdään, ja poistu sen jälkeen ulompaan
        koodiin break:n avulla
14 }
```

C. Vastusten värikoodit

Arduinon mukana tulevissa vastuksissa on käytössä joko kahdenlaisia vastusmerkintöjä, joko neljällä tai viidellä viivalla.

Näistä viimeinen kertoo kuinka tarkasti vastuksen arvo osuu annettuun arvoon, ja nämä viivat ovat usien kultaisia tai hopeisia. Tämän avulla voit yrittää tunnistaa kummasta päästä vastuksen arvoa ollaan lukemassa.

Kaksi tai kolme ensimmäistä viivaa kuvaa vastuksen numeroarvoa ja kolmas tai neljäs viiva kuvaa vastuksen kokoluokkaa (eli kerrointa, esimerkiksi $k = 1000$).

Kahden tai kolmen ensimmäisen viivan
värit ja niitä vastaavat numerot:

0
1
2
3
4
5
6
7
0
9

Kertoimen suuruus, 10^x , missä x on listassa annettu luku:

0
1
2
3
4
5
6

Toleranssin
suuruus:

$\pm 1\%$
$\pm 2\%$
$\pm 5\%$ kulta
$\pm 10\%$ hopea

HUOM!

Jos käytössäsi on yleismittari, voit tarkistaa vastuksen arvon sillä!

Taulukko C.1: Paketissa mukana olevat vastukset.

Vastus	Kuvat
220Ω	 / 
560Ω	 / 
1kΩ	 / 
4.7kΩ	 / 
10kΩ	 / 
1MΩ	 / 
10MΩ	 / 

Arduino-paketti yläkouluun ja lukioon

- Koekytkentälevy
- LED
- Liikennevalot
- Lämpö- ja valoanturit
- LCD-näyttö
- Kaltevuusanturi

