

Arduinin ohjelmointi

Sami-Petteri Pukkila

sami-petteri.pukkila@aalto.fi

6. helmikuuta 2017

Mistä kaikki lähtee liikkeelle?

- ▶ Ennen kuin alat ohjelmoimaan, mieti mitä projektissa kannattaa tehdä.
- ▶ Jaa projekti pieniin osaprojekteihin ja kehitä näille ratkaisu jonka voi myöhemmin yhdistää kokonaisuuteen.

Esimerkki esteitä väistelevä auto

- ▶ Miten auto liikkuu?
- ▶ Miten tunnistan esteen?
- ▶ Mitä teen kun tunnistan esteen?

Esimerkki esteitä väistelevä auto

- ▶ Miten auto liikkuu?
 - ▶ Tarvitaan kai moottoreita. Miten niitä ohjataan?
- ▶ Miten tunnistan esteen?
 - ▶ Jonkinlainen etäisyysanturi. Miten niitä luetaan?
- ▶ Mitä teen kun tunnistan esteen?
 - ▶ Varmaankin käännyn kunnes este häviää näkökentästä?

Esimerkki esteitä väistelevä auto

- ▶ Miten auto liikkuu?
 - ▶ Tarvitaan kai moottoreita. Miten niitä ohjataan?
 - ▶ Voisin kokeilla ajaa moottoreita moottorinohjaimella.
 - ▶ Voisin tehdä funktion jolle kerrotaan mitä moottoria pitäisi pyörittää millä nopeudella ja mihin suuntaan
- ▶ Miten tunnistan esteen?
 - ▶ Jonkinlainen etäisyysanturi. Miten niitä luetaan?
 - ▶ Löysin ultraäänianturin joka kertoo etäisyyden seuraavaan esteeseen.
 - ▶ Voisin tehdä funktion joka palauttaa etäisyyden seuraavaan esteeseen.
- ▶ Mitä teen kun tunnistan esteen?
 - ▶ Voinkin kätevästi yhdistää kaksi jo tehtyä funktiota
- ▶ Tarvitsenkohan etäohjauksen...

Koodin laatu

- ▶ Joissain osaongelmissa vierähtää pidempi aika, joissakin lyhyempi
- ▶ Olisi kuitenkin kiva jos muutama viikko sitten tehty koodi ei olisi täyttä mysteeriä
- ▶ Kirjoita mielummin yksinkertaista ja "typerää" koodia kuin kikkailuja sisältävää "fiksua" koodia

Kommentit

- ▶ Koodin sisälle voi kirjoittaa kommenttirivejä
- ▶ Kommentti voi olla esimerkiksi selostus, mitä funktio tekee, mitä se palauttaa ja mitä se ottaa sisään
- ▶ Jos luulet unohtavasti miksi olet tehnyt jonkin asian kuten olet, kannattaa siitä kirjoittaa kommentti
- ▶ Arduinossa kommenttirivi aloitetaan kahdella kauttaviivalla (*// kommentti*)

Sisennys

- ▶ Sisennetty koodi on helppolukuista sillä lohkot erottuvat visuaalisesti toisistaan
- ▶ Koodia saa sisennettyä painamalla tab-nappulaa kursorin ollessa rivin alussa

```
void loop()
{
String nimi;
if (tila == tila1)
nimi = kysy_nimi();
if (tila == tila2)
sano_moi(nimi);
}
```

```
void loop()
{
    String nimi;

    if (tila == tila1)
        nimi = kysy_nimi();
    if (tila == tila2)
        sano_moi(nimi);
}
```


Arduinon kieli

<https://www.arduino.cc/en/Reference/HomePage>

Arduinon kieli on käytännössä C/C++:aa

- ▶ Kaikki muistakin ohjelmointikielistä tutut rakenteet löytyvät
 - ▶ Muuttujat, funktiot
 - ▶ silmukat, laskutoimitukset, ehtolauseet
- ▶ Kieli on laiteläheinen ja käännetään ennen lähettämistä binääriksi.
 - ▶ Osa asioista vaikuttaa kankeammilta kuin esim. Pythonissa

Mistä löytyy tietoa?

- ▶ Arduinon referenssi:
<https://www.arduino.cc/en/Reference/HomePage>
- ▶ Arduinon esimerkit:
<https://www.arduino.cc/en/Tutorial/BuiltInExamples>
- ▶ Google: *"How to do x in arduino"* *"How x works in arduino"*
 - ▶ *"How to do x in c"*

Muuttujat

<https://www.arduino.cc/en/Tutorial/Variables>

`int lampotila = 5 ;`
tyyppi nimi arvo

- ▶ Muuttujalla on oltava tyyppi (kokonaisluku, kirjan...)
- ▶ Eroaa esimerkiksi Pythonista jossa näin ei ole!
- ▶ Määrää mitä muuttuja voi pitää sisällään, mitä ei.

Funktiot

```
int funktionimi(int parametri1, int parametri2)
{
    return paluuarvo;
}
```

Funktio ottaa sisäänsä parametreja, tekee niillä jotakin ja antaa jonkinlaisen paluuarvon.

- ▶ **Paluutyyppi**
 - ▶ Muuttujan palauttaman arvon tyyppi.
- ▶ **Funktionimi**
 - ▶ Pitää alkaa kirjaimella.
- ▶ **Parametri**
 - ▶ Parametrit joita funktio ottaa vastaan. Voi olla useampia ja jokaisella pitää olla tyyppi.
 - ▶ Pilkillä voi erottaa useampia parametreja.

Taulukot

- ▶ Hyvä väline esimerkiksi useamman ledin nastanumeroiden säilömiseen.

```
int led_pin[3] = {3, 5, 6};  
//led_pin[0] on nyt 3  
//led_pin[1] vastaavasti 5
```

- ▶ Taulukon jäseniä kutsutaan alkioksi.
- ▶ Alkioon pääsee käsiksi syntaksilla `nimi[alkio]`
- ▶ Indeksointi alkaa nolasta

```
led_pin[1] = 2;  
//led_pin vastaa nyt määritelmää {3, 2, 6}
```

Teksti

<https://www.arduino.cc/en/Reference/String>

- ▶ C-kielessä ei ole erillistä muuttujatyyppiä tekstille, vain merkeille (char)
- ▶ Merkkijonot esitetään char-taulukkona

```
char materiaali [] = "puu";  
// {'p', 'u', 'u', '\0'}
```

- ▶ Merkkijonon lopussa ns. nollatavu
 - ▶ Työkalut pystyvät päättelemään mihin merkkijono loppuu.
 - ▶ Kolmen merkin merkkijonon tallentamiseen tarvitaan siis neljä merkkiä!

String-objekti

<https://www.arduino.cc/en/Reference/StringObject>

- ▶ C-kielessä tekstinkäsittely on hieman kankeaa
- ▶ Arduinossa on olemassa String-objekti joka tekee tästä hieman helpompaa
 - ▶ Käyttäkää sitä jos C ei ole tuttu

Muuttujien näkyvyys

```
int a = 1;

void setup()
{
  int b = a + 2; //Toimii
}

void loop()
{
  a = b + 2; //Ei toimi, b näkyy
             //vain setup-funktiossa
}
```


Globaali vs lokaali muuttuja

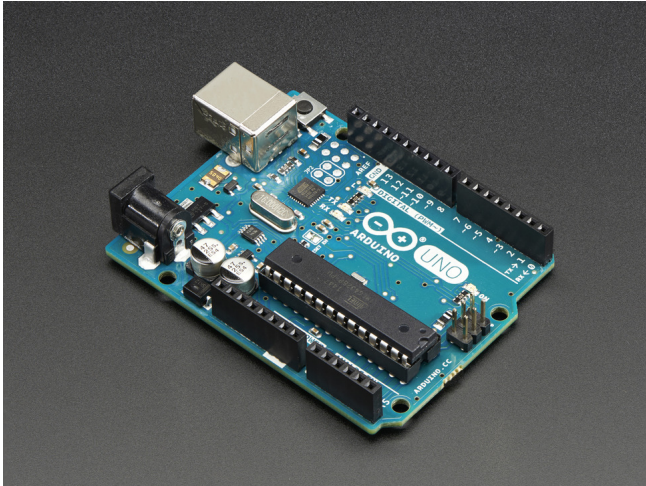
Globaali (esimerkissä a)

- ▶ Voidaan käyttää kaikkialla ohjelmassa
- ▶ Hyviä kun samaa muuttujaa käytetään kaikkialla ohjelmassa. Esim. Arduinoon kytketyn ledin pinninumero
- ▶ **Vältä mikäli mahdollista.** Hankala tutkia mitkä kaikki ohjelman osat vaikuttavat muuttujaan.

Lokaali (esimerkissä b)

- ▶ Voi käyttää vain niiden alaosuhteiden välissä jossa määritetty
- ▶ Laitteen tarvitsee säilöä tieto ainoastaan pienessä osassa ohjelmaa → tehokkaampaa
- ▶ Vaikeampi tehdä mokia

Arduino



Arduino-ohjelman rakenne

```
int ledpin = 13;

void setup()
{
  pinMode(ledpin, OUTPUT);
}

void loop() {
  digitalWrite(ledpin, HIGH);
  delay(1000);
  digitalWrite(ledpin, LOW);
  delay(1000);
}
```

Arduino-ohjelman rakenne

- ▶ **setup-funktio**
 - ▶ Ajetaan kerran kun Arduinoon kytketään virrat
- ▶ **loop-funktio**
 - ▶ Setup-funktion jälkeen ajetaan uudelleen ja uudelleen kunnes Arduinosta katkaistaan virrat

digitalWrite

<https://www.arduino.cc/en/Tutorial/DigitalPins>

- ▶ Asettaa pinnin arvon LOW tai HIGH.
 - ▶ LOW, "pois päältä"
 - ▶ HIGH, "päällä"

Esimerkiksi ledin vilkutus

```
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH); //LED päällä
  delay(1000);           //Odotetaan sekunti
  digitalWrite(13, LOW); //LED pois päältä
  delay(1000);           //Odotetaan sekunti
}
```

digitalRead

- ▶ Lukee pinnin tilan
 - ▶ **LOW**, pinni kytketty maahan (GND)
 - ▶ **HIGH**, pinni kytketty jännitteeseen. (3-5 voltia)

```
void setup()
{
  pinMode(3, INPUT);
}

void loop()
{
  int arvo = digitalRead(3);
}
```

Ylös-/Alasvetovastukset

<https://learn.sparkfun.com/tutorials/pull-up-resistors>

- ▶ Mikäli pinniä ei ole kytketty mihinkään, sanotaan että se kelluu
 - ▶ digitalRead palauttaa satunnaisesti HIGH tai LOW
- ▶ Ratkaisuna **al**asvetovastus
 - ▶ Kytketään pinni vastuksella (n. 10K Ω) **GND**:hen
 - ▶ Pinnin arvo on nyt **LOW** kunnes se kytketään myös +5V:iin
- ▶ **Yl**ös vetovastus lähes sama asia:
 - ▶ Kytketään pinni vastuksella (n. 10K Ω) **+5V**:iin
 - ▶ Pinnin arvo on nyt **HIGH** kunnes se kytketään myös GND:iin

INPUT_PULLUP

- ▶ Arduinosta löytyy sisäänrakennettuna ylösvetovastus
- ▶ Aktivoidaan määrittämällä pinnin modeksi **INPUT_PULLUP**
- ▶ Yksinkertainen nappula voidaan nyt kytkeä suoraan GND:in ja halutun pinnin välille. Mitään ylimääräistä vastusta ei tarvita kytkennässä!

```
void setup()
{
  pinMode(3, INPUT_PULLUP);
}

void loop()
{
  int nappula = digitalRead(3);
}
```


analogRead

- ▶ analogReadilla voidaan lukea pinnin jännite
- ▶ Onnistuu vain pinneistä jotka on nimetty A0, A1...
 - ▶ Kyseisten pinnien takana on 10-bittinen analogidigitaalimuuntaja (AD)
 - ▶ Arvo 0-5V väliltä saadaan lukuna väliltä 0-1023. (2,5V siis vastaisi lukua 512)
 - ▶ Kyseisiä pinnejä voi käyttää myös digitaaliseen I/O:hon

```
void setup()
{
  pinMode(A1, INPUT);
}

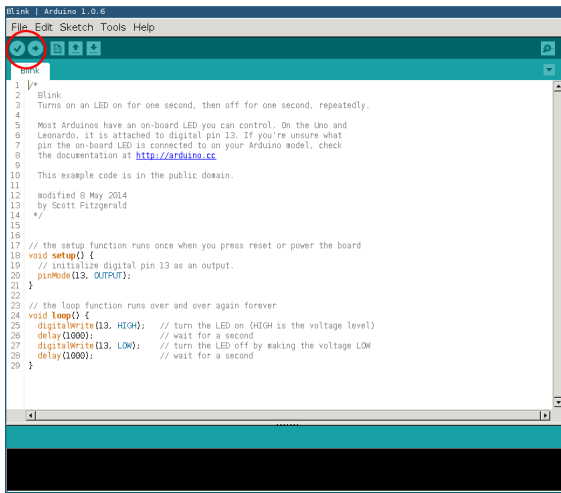
void loop()
{
  int arvo = analogRead(A1);
}
```

analogWrite

- ▶ Arduinossa ei ole DA-muunninta → analoginen ulostulo pitää tehdä "keinotekoisesti"
 - ▶ PWM, eli Arduino kytkee pinniä nopeasti 0V ja 5V välillä. (kanttiaalto)
 - ▶ "Teho" määräytyy kauanko pinniä pidetään HIGH- ja kauanko LOW-tilassa
 - ▶ Esimerkiksi LEDien himmentämisessä ja normaalien moottorien tehon säädössä riittävä
 - ▶ Joissain uusissa Arduinoissa tosin on DA-muunnin.
- ▶ Vain tildellä (~) merkityjä pinnejä voi käyttää

```
void setup() {  
  pinMode(3, OUTPUT);  
}  
void loop() { //Ledi palaa "puolella teholla"  
  analogWrite(3, 127);  
}
```

Jos ohjelma ei käänny



```
Arduino 1.0.6
File Edit Sketch Tools Help
Blink
1 /*
2  * Blink
3  * Turns on an LED on for one second, then off for one second, repeatedly.
4  *
5  * Most Arduinos have an on-board LED you can control. On the I/O and
6  * Leonardo, it is attached to digital pin 13. If you're unsure what
7  * pin the on-board LED is connected to on your Arduino model, check
8  * the documentation at http://arduino.cc
9  *
10 * This example code is in the public domain.
11 *
12 * modified 8 May 2014
13 * by Scott Fitzgerald
14 */
15
16
17 // the setup function runs once when you press reset or power the board
18 void setup() {
19   // initialize digital pin 13 as an output.
20   pinMode(13, OUTPUT);
21 }
22
23 // the loop function runs over and over again forever
24 void loop() {
25   digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
26   delay(1000); // wait for a second
27   digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
28   delay(1000); // wait for a second
29 }
```

Jos ohjelma ei käänny

- ▶ Arduino yleensä kertoo aika hyvän vihjeen mikä ongelma koodissa on
- ▶ Jos vihje vaikuttaa mystiseltä, copy&pastea se Googleen. Yleensä joku muu on tehnyt saman virheen aikaisemmin.

Kun ohjelma ei toimi kuten oletin

- ▶ Monesti auttaa kun ohjelman logiikkaa lukee mielessään
- ▶ *"Kerro lämpötila käyttäjälle, tämän jälkeen lue lämpötila sensorilta..."* Ehkä toisin päin
- ▶ Kumiankkadebuggaus

Helpot asiat mokata: sijoitus vs vertailu

- ▶ **a=b ja a==b ovat eri asioita!**
 - ▶ a = b; sijoita b:n arvo a:han
 - ▶ a == b; tarkista ovatko a ja b saman arvoiset
 - ▶ *if (a = b)* on siis useimmiten virhe!
- ▶ C-tyylisiä merkkijonoja ei voi vertailla ==-operaattorilla. Arduinon String-olioita voi!

Muuttujatyypit laskutoimituksissa

Arduino ei automaattisesti muunna muuttujatyyppejä ihmisen mieleen sopiviksi.

```
int kokonaisluku = 2;
float muuttuja1 = 3/kokonaisluku;           // 1
float muuttuja2 = 3.0/2.0;                  // 1.5
float muuttuja3 = 3.0/(float)kokonaisluku; // 1.5
```

- ▶ (tyyppi)muuttuja-notaatio muuntaa muuttujan tyyppin
- ▶ Muunnos isommasta tyypestä pienempään aiheuttaa ongelmia jos muunnettava arvo on liian iso. (long → int)

Under-/overflow

- ▶ Mikäli muuttujan arvo menee muuttujatyyppin alueen yli, "pyörähtää muuttuja yli"
- ▶ Byte-tyypin maksimiarvo on 255.
- ▶ Huomaa ilmiö erityisesti millis()-funktion käytössä

```
byte c = 0; // c = 0
--c;      // c = 255
c = c + 1; // c = 0
```

Tähän käyttäytymismalliin ei kannata luottaa, ei ole standardissa.

Sarjaportti

<https://www.arduino.cc/en/Reference/Serial>

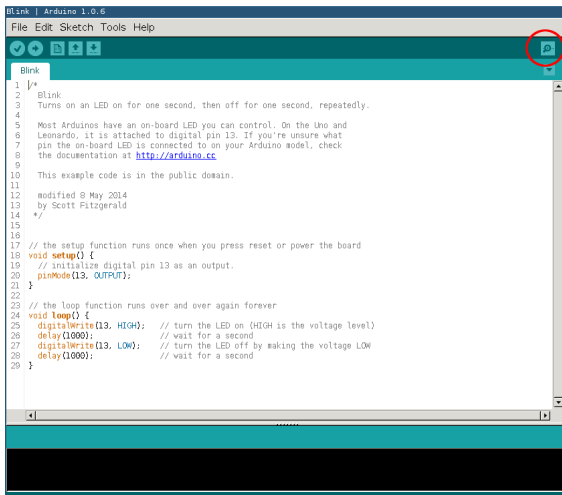
- ▶ Arduino osaa USB:n kautta esittää sarjaporttia
- ▶ Helpoin tapa tietää mitä Arduinossa tapahtuu
- ▶ Kun ohjelma ei toimi kuten sen pitäisi, kannattaa sarjaporttiin tulostaa eri muuttujien arvoja ja tutkia niiden järkevyyttä
- ▶ Tekstin tulostaminen ennen ja jälkeen jonkin ohjelman osan auttaa selvittämään, jääkö ohjelma jumiin johonkin kohtaan.

Sarjaportti

<https://www.arduino.cc/en/Reference/Serial>

- ▶ Arduino osaa USB:n kautta esittää sarjaporttia
- ▶ Helpoin tapa tietää mitä Arduinossa tapahtuu
- ▶ Kun ohjelma ei toimi kuten sen pitäisi, kannattaa sarjaporttiin tulostaa eri muuttujien arvoja ja tutkia niiden järkevyyttä
- ▶ Tekstin tulostaminen ennen ja jälkeen jonkin ohjelman osan auttaa selvittämään, jääkö ohjelma jumiin johonkin kohtaan.
- ▶ **Huom!** jos käytät sarjaporttia, pinnejä 0 ja 1 ei voi enää käyttää muuhun! (UNOssa)
- ▶ Voi käyttää viestintään myös Arduinon suuntaan tai useamman Arduinon välillä

Serial monitorin avaaminen



Varmista että tools → port on valittu oikein.

Linkkejä

- ▶ Arduinon oppaat ja esimerkit
 - ▶ <https://www.arduino.cc/en/Tutorial/HomePage>
- ▶ Arduinon kielen referenssi
 - ▶ <https://www.arduino.cc/en/Reference/HomePage>
- ▶ Google
 - ▶ <http://www.google.com>