

Muisteluja Arduinon alkeiskurssien demoista, **saa muokata ja lisäillä kysymyksiä.**

(Saa myös päivittää, tällä alkaa jo olla ikää)

0. Asennus

- Arduino IDE asennettuna
- File ► Examples ► 01.Basics ► Blink
 - testattu ohjelmointi
 - Arduino UNO:lla standard USB serial driver
 - Arduino Nano:lla tai Duemilanovella FTDI serial driver
 - Linuxit saattaa vaatia, että käyttäjä lisätään dialup-grouppiin ja kirjaudutaan uudestaan
 - Macissä taitaa toimia aina suoraan (pitää tarkistaa FTDI ongelma!)
 - Windowseissa riippuu kaikesta, tähän pitää ensi kerralla kerätä eri versioiden asennus uusimmalla Arduino releasella

1. LEDien vilkuttelua

- Muokataan Blink-esimerkkiä siten, että ledi on päällä puoli sekunttia ja pois päältä kaksi sekunttia
- Tallennetaan esimerkki Save As:llä eri nimisenä omaan projektikansioon
 - kerrotaan että Upload ohjelmoin, mutta ei tallenna tiedostoa ja että IDE on todella helppo saada täysin jumiin ja viimeiset muutokset katoavat jos ei muista välillä tallentaa
 - kerrotaan että Arduino-projektit koostuvat kansioista ja saman nimisestä .ino (vanhat projektit .pde) tiedostosta sekä mahdollisesti muista .h yms. lisätiedostoista
 - projektin nimessä ei saa olla erikoismerkkejä, välilyöntejä eikä se saa alkaa numerolla
 - sen pitää täyttää Java-ohjelmointikielen luokan nimen vaatimukset, koska IDE on tehty Javalla (todo: pitää tarkistaa mikä tilanne nykyään)
 - Arduino 'kieli' on itse asiassa syntaksiltaan C++:aa, ilman standardikirjastoja, mutta siitä nyt ei alkeiskurssilla paljon tarvi välittää
- Tässä pitäisi puhua LEDeistä pitkään.
- Seuraavana opetellaan leipälaudan käyttöä, painotetaan etuvastuksen käyttöä aina kun ledejä kytketään mihinkään, sopivat arvot saa netistä löytyvillä laskimilla ja sen tulisi olla jotain 100-400 ohmin välillä
- muokataan ohjelmaa siten, että se käyttää ulkoista lediä pinnin 10 kautta, (sen lisäksi, että 13 on käytössä; jos aikaa riittää, niin laitetaan vilkkumaan eri tahtiin)
- muokataan ohjelmaa siten, että kokeillaan (jos käytössä skooppi demotaan myös)
 - `analogWrite(10, 52);`
 - `analogWrite(10, 178);`
 - `analogWrite(10, 255);`
- kerrotaan, että `analogWrite` oikeasti on digitaalinen PWM ellei sitä esim. kondensaattorilla tasoita, mutta ledien ohjaamiseen se on optimaalinen
- (todo: lisää kunhan muistiin tulee)
- Jatkotehtävä: 'pumppaa' lediä 13 pehmeästi
 - `// Ohjelmallinen PWM`
 - `const int ledin_pinni = 13;`
 -
 - `void setup() {`
 - `pinMode(ledin_pinni, OUTPUT);`
 - `}`

-
- void loop() {
- for (int kirkkaus = 0; kirkkaus < 256; kirkkaus++) {
- for (int toisto = 0; toisto < 30; toisto++) {
- digitalWrite(ledin_pinni, HIGH);
- delayMicroseconds(kirkkaus + 1);
- digitalWrite(ledin_pinni, LOW);
- delayMicroseconds((255 - kirkkaus) + 1);
- }
- }
- for (int kirkkaus = 255; kirkkaus >= 0; kirkkaus--) {
- for (int toisto = 0; toisto < 30; toisto++) {
- digitalWrite(ledin_pinni, HIGH);
- delayMicroseconds(kirkkaus + 1);
- digitalWrite(ledin_pinni, LOW);
- delayMicroseconds((255 - kirkkaus) + 1);
- }
- }
- }
- }

2. Analogisen jännitteen mittaamista

- Otetaan pohjaksi File ► Examples ► 03.Analog ► AnalogInOutSerial
 - kytketään potentiometrin päät Vcc ja Gnd ja keskipiikki A0
 - kytketään ukoinen ledi etuvastuksella pinniin 9
 - katsotaan Serial Monitorilla arvoja
 - nostetaan baudirate 115200 ohjelmassa ja monitorissa ja jatkossa käytetään aina tätä koska muuten suotta hidastetaan Arduinon toimintaa
 - demotaan miltä näyttää jos baudiratat eivät ole samat tietokoneen päässä kuin millä Arduino dataa kirjoittaa
 - muistetaan aina laittaa ohjelmaan rivinvaihto ja delay jos se kirjoittaa jotain serialiin
 - opetellaan muuttamaan ohjelmaa siten, että potentiometri toimii käänteisesti
 - vihje: (255 - outputValue)
 - tätä tullaan tarvitsemaan RGB-ledien kanssa koska niistä monet ovat common anode -tyyppisiä

Tästä kohdasta puuttuu jotain.

koodista / syntaksista tai muuttujista en sano mitään.

En sano myöskään paljoa siitä, että se FOR-silmukka ei nyt vaan ole helpposti omaksuttava juttu jos sitä ei saa luntata jostain. (asiana se on ymmärrettävissä, mutta se kirjoittaminen ei ole ihan itsestään selvää)

2.5 kynnys vaikka kolmella LEDillä. (jos mahtuisi päivän ohjelmaan)

```
void loop() {
  // read the sensor:
  int sensorReading = analogRead(A0);
  // map the sensor range to a range of three options:
  int range = map(sensorReading, sensorMin, sensorMax, 0, 2);
  .....
```

3. RGB-ledin PWM ohjausta

- ensin testataan etuvastukset eri värikomponenteille jotta saadaan aikaan valkoinen kun ledi on kytketty Vcc ja Gnd ja samalla selviää onko se common anode vai common catode tyyppinen
- Harjoitellaan kirjottamaan sanelun mukaan ohjelma tyhjästä
 - alussa määritetään kolme vakiomuuttujaa jotka määrittelevät ledin pinninumerot, valitaan vapaasti kuudesta PWM:n osaavasta pinnistä
 - setupissa määritetään kyseiset pinnit ulostuloiksi
 - loopissa tehdään kuusi for-simukkaa, kaksi joka värille joissa
 - ensin punaisen pwm arvo menee 0:sta 255:een ja joka arvoa pidetään 5 millisekuntia
 - sitten arvo laitetaan 255:sta 0:aan vastaavasti
 - vastaavat kaksi silmukkaa vihreälle ja siniselle ledille
 - silmukoiden jälkeen sekunnin delay, jotta huomataan jos ledin värit väärässä järjestyksessä
 - tämän jälkeen muutellaan silmukoiden järjestystä ja ihmetellään loputulosta

Tässä sitä 'sanelua'

```
const int punaisenLedinPinni = 11;  
const int vihreanLedinPinni = 10;  
const int sinisenLedinPinni = 9;
```

```
void setup() {  
  pinMode(punaisenLedinPinni, OUTPUT);  
  pinMode(vihreanLedinPinni, OUTPUT);  
  pinMode(sinisenLedinPinni, OUTPUT);  
}
```

```
void loop() {  
  // for-simukassa ensin kerrotaan ns. silmukkamuuttuja ja annetaan sille alkuarvo  
  // sitten seuraavana tulee ehto että kuinka kauan silmukkaa pyöritetään  
  // ja viimeisenä muutetaan muuttujaa sopivasti, tässä käyty ++ koska se on todella  
  // yleinen C-kielen tyylisissä ohjelmointikielissä lyhennys tästä:  
  // punaisenArvo = punaisenArvo + 1  
  // ja vertailussa käytetty <= tarkoittaa pienempi tai yhtäsuuri kuin  
  // sitten aaltosulkujen sisältö suoritetaan aina joka kierroksella ja silmukkamuuttuja  
  // käytettävissä, ei tietysti pakko jos ei tarvetta  
  // eli tässä se saa arvot 0, 1, ..., 254 ja 255  
  for (int punaisenArvo = 0; punaisenArvo <= 255; punaisenArvo++) {  
    analogWrite(punaisenLedinPinni , punaisenArvo);  
    delay(5);  
  }  
  // Tässä vastaavasti 255, 254, ..., 1, 0  
  for (int punaisenArvo = 255; punaisenArvo >= 0; punaisenArvo--) {  
    analogWrite(punaisenLedinPinni , punaisenArvo);  
    delay(5);  
  }  
  
  for (int vihreanArvo = 0; vihreanArvo <= 255; vihreanArvo++) {  
    analogWrite(vihreanLedinPinni , vihreanArvo);  
    delay(5);  
  }
```

```

}
for (int vihreanArvo = 255; vihreanArvo >= 0; vihreanArvo--) {
  analogWrite(vihreanLedinPinni , vihreanArvo);
  delay(5);
}

for (int sinisenArvo = 0; sinisenArvo <= 255; sinisenArvo++) {
  analogWrite(sinisenLedinPinni , sinisenArvo);
  delay(5);
}
for (int sinisenArvo = 255; sinisenArvo >= 0; sinisenArvo--) {
  analogWrite(sinisenLedinPinni , sinisenArvo);
  delay(5);
}

delay(1000);

}

```

4. Servon ohjausta potikalla

- Nyt pitäisi omin neuvoin saada toimimaan File ► Examples ► Servo ► Knob
 - etsitään vastaava tutoriaali <http://arduino.cc/en/Tutorial/HomePage>

5. Muiden analogisten sensoreiden lukeminen

- kirjoitetaan seuraava ohjelma ja testataan se potentiometrillä
- sitten kytketään 'vastusjakoon' fototransistori, fotoresistori tai joku vastaava ja toiseksi joku kiinteä vastus 1k, 10k, 100k tai 1M ohmin joukosta siten että sensoria aktivoimalla saadaan mahdollisimman suuri arvojen vaihtelu

ohjelma 5 alkaa tästä ->

// Tällä voidaan helpommin visualisoida analogisten antureiden arvoalueita. Testaa ensin potentiometrillä.

```

void setup() {
  Serial.begin(115200);
}

// Tulostaa luvun 0..1023 aina neljän merkin mittaisena " 0", " 99", .. " 999", "1023"
void tulosta_arvo(int analogi_arvo) {
  if (analogi_arvo < 10) {
    Serial.print(" ");
  } else if (analogi_arvo < 100) {
    Serial.print(" ");
  } else if (analogi_arvo < 1000) {
    Serial.print(" ");
  }

  Serial.print(analogi_arvo);
}

```

// Palkin pituus pitää säätää oman näytön mukaan siten, että Serial Monitorissa rivit mahtuvat ruudulle, mutta ovat mahdollisimman pitkiä

```
void piirra_palkki(int analogi_arvo, int palkin_pituus = 80) {  
  // Skaalataan arvo 0..1023 piirtoa varten 0..palkin_pituus  
  int palkin_arvo = (long(analogi_arvo)*palkin_pituus)/1023;
```

// Piirretään palkki "=====>....." tyylisesti, tätä koodia ei tarvitse ymmärtää

```
for (int i = 0; i < palkin_pituus; ++i) {  
  if (palkin_arvo > i) {  
    Serial.print("=");  
  } else if (palkin_arvo == i) {  
    Serial.print(">");  
  } else if (i % 10 == 0) {  
    Serial.write(0xb7); // "."  
  } else {  
    Serial.write("."); // "."  
  }  
}  
}
```

```
void loop() {  
  // Luetaan analogi arvo  
  int arvo = analogRead(A0);
```

// Tulostetaan arvo tasaleveisenä lukuna
tulosta_arvo(arvo);

// Piirretään perään palkkina, säädä oman ruudun leveyden ja Serial Monitorin mukaan
piirra_palkki(arvo, 80);

// Muistetaan rivinvaihto, tai muuten Serial Monitor menee jumiin
Serial.println();

```
// Tällä voidaan hidastaa piirtoa  
delay(10);  
}
```

<-- ohjelma 5 loppuu tähän

-- kilpurille :

```
const int punaisenLedinPinni = 11;  
const int vihreanLedinPinni = 10;  
const int sinisenLedinPinni = 9;
```

```
const int punaisenArvoPantterille = 255; // punainen saa olla täysillä  
const int vihreanArvoPantterille = 0;    // vihreätä ei pantteriin tarvita  
const int sinisenArvoPantterille = 100;  // tätä arvoa säädät
```

```
void setup() {  
  pinMode(punaisenLedinPinni, OUTPUT);  
  pinMode(vihreanLedinPinni, OUTPUT);  
  pinMode(sinisenLedinPinni, OUTPUT);  
  
  analogWrite(punaisenLedinPinni, punaisenArvoPantterille);  
  analogWrite(vihreanLedinPinni, vihreanArvoPantterille);  
  analogWrite(sinisenLedinPinni, sinisenArvoPantterille);  
}
```

```
void loop() {  
  // loopissa ei tarvitse tässä tehdä mitään, koska Arduinon rauta hoitaa PWM:t taustalla  
}
```

----- MultiBlink -----

// Vilkutetaan kahta lediä täysin toisistaan riippumatta, tätä ei delay:llä pysty tekemään

```
void setup() {  
  pinMode(12, OUTPUT);  
  pinMode(11, OUTPUT);  
}
```

```
const int viive1 = 300;  
const int viive2 = 700;
```

```
long aika1 = 0;  
long aika2 = 0;
```

// the loop function runs over and over again forever

```
void loop() {  
  if (millis() - aika1 > viive1) {  
    aika1 = millis();  
    digitalWrite(11, !digitalRead(11));  
  }
```

```
  if (millis() - aika2 > viive2) {
```

```
aika2 = millis();  
digitalWrite(12, !digitalRead(12));  
}  
}
```