



ELEKTRONIIKKAOHJELMOINNIN PERUSTEET

ARDUINO JA MICRO:BIT



SISÄLTÖ: LAITTEISTO

Ohjelmoitavat kehitysalustat

- Arduino
- micro:bit
- StarterKits

Virtapiirit

- Virtapiiri ja Ohmin laki
- Vs+ ja GND
- Koekytkentälevyt
- Breakout board micro:bitille
- Vastukset
- Vastuksen resistanssin lukeminen

Input-komponentteja:

- Potentiometri
- Valovastus
- Lämpötilasensori
- Painokytkin
- Asentokytkin

- IR-vastaanotin (ja –lähetin)
- Ultraäänisensori

Output-komponentteja:

- LED ja RGB-LED
- Pietsosummeri
- Servomoottori
- DC-moottori
- LED-näyttö

Muita hyödyllisiä komponentteja:

- Kondensaattori
- Rele
- Transistori
- Diodi
- H-silta

SISÄLTÖ: OHJELMOINTI

Ohjelmointiympäristöjä

- Arduino
 - [Tinkercad](#)
 - [ArduinoCC](#)
- Micro:bit
 - [JS Blocks](#)
 - [Mbed](#)



Ohjelmoinnin peruselementtejä

- Muuttujat
- Ehtolauseet
- Silmukat
- Funktiot
- Kirjastot
- Oliot



OHJELMOITAVAT KEHITYSALUSTAT

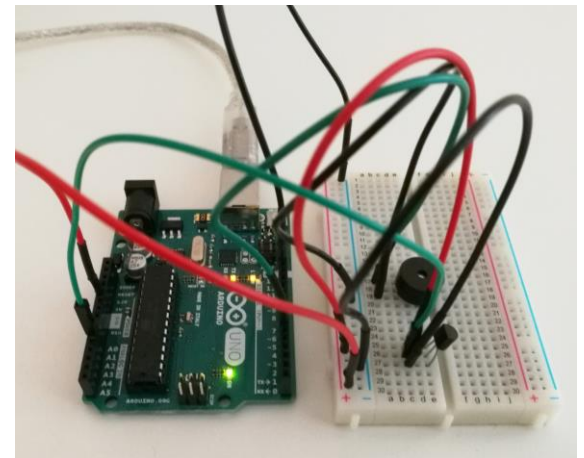
JOHDANTO

- Arduino sekä micro:bit kehitysalustat sopivat hyvin ohjelmoinnin ja elektroniikan alkeiden opettamiseen
- Ne vastaavat toiminnoiltaan mikrokontrollereita, jotka ovat pieniä ohjelmoitavia tietokoneita, joita käytetään usein sulautetuissa järjestelmissä erilaisten laitteiden ohjaamiseen (esim. pesukoneet, mikroaaltouunit, varashälyttimet jne.)
- Yhteistä näille laitteille on se, että niissä voi toimia kerrallaan vain yksi ohjelma
- Kehitysalustat koostuvat prosessorin ja muistin lisäksi erilaisista liitännöistä esim. virtalähteelle, input/output-komponenteille sekä mikrokontrollerin ohjelmointia varten
- Kehitysalustoja ohjelmoidaan tietokoneen avulla ja tehty ohjelma ladataan siihen USB-johdon avulla (micro:bit:ia voidaan ohjelmoida myös iPadillä Bluetooth yhteyden kautta)
- Laitteeseen ladattu ohjelma toimii siinä niin kauan kun se saa virtaa

OHJELMOITAVAT KEHITYSALUSTAT

JOHDANTO

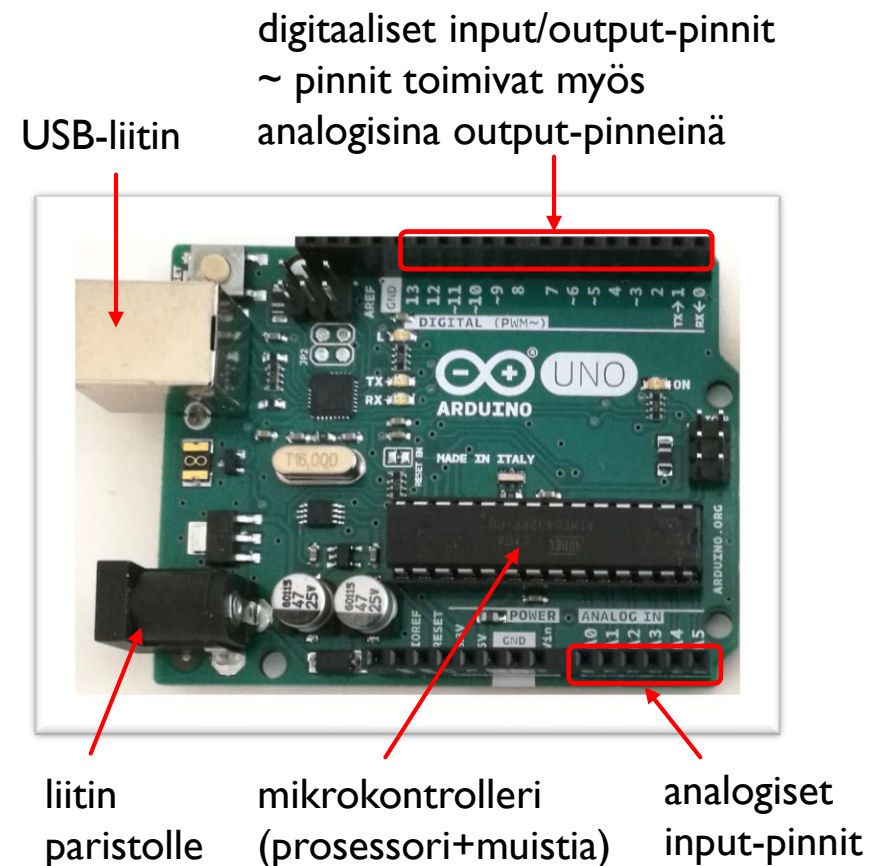
- Kehitysalustat saavat virtaa USB-johdon kautta tai erillisestä paristosta
- Laitteisto saa tietoa ympäristöstään joko kehitysalustalle kiinteästi integroitujen tai sen pinneihin liitettyjen *input-komponenttien* avulla.
- Laitteisto voi vuorovaikuttaa ympäristönsä kanssa joko kehitysalustalle kiinteästi integroitujen tai sen pinneihin kiinnitettyjen *output-komponenttien* avulla
- Ulkoiset input/output-komponentit liitetään useimmiten mikropiirin GPIO-pinneihin (*General Purpose Input Output*)
- Näiden laitteiden ohjelmointi on yksinkertaisimmillaan näiden GPIO-pinnien lukemista ja kirjoittamista



Arduinon input/output pinneihin on kiinnitetty lämpötilasensori ja pietsosummeri

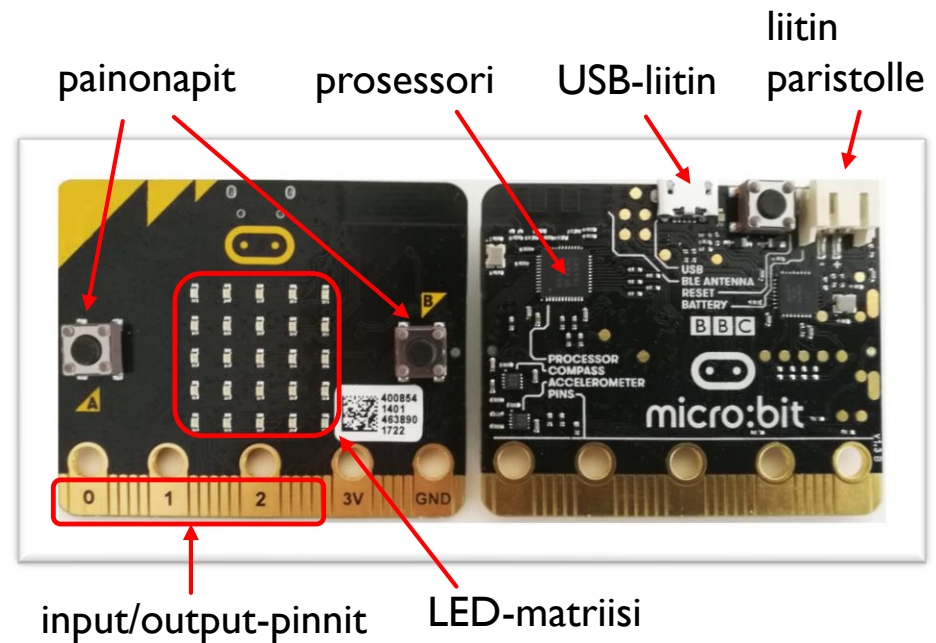
ARDUINO

- Arduino on avoimeen laitteistoon perustuva kehitysalusta, jonka ”aivoina” toimii mikrokontrolleri (8-bittinen Atmel AVR). Arduinoja on useita erilaisia, tässä käytämme Arduino UNO-alustaa.
- Arduino UNO:ssa on
 - 6 analogista input-pinniä
 - 8 digitaalista input/output-pinniä sekä
 - 6 pinniä, jotka voivat toimia sekä digitaalisina input/output-pinneinä, että analogisina output pinneinä (~ merkintä)
- Ohjelmointi tapahtuu USB-johdon avulla
- Laitteisto vaatii toimiakseen 9V jännitteen (USB-johdon kautta tai paristolla) ja se antaa ulos maksimissaan 5V



MICRO:BIT

- micro:bit on BBC:n kehittämä kehitysalusta, jonka aivoina toimii ARM Cortex-M0 pohjainen prosessori
- micro:bit:issä on
 - 3 leveää input/output-pinniä, joita voidaan käyttää sekä digitaalisina että analogisina
 - 14 kapeaa input/output-pinniä, joista 3 toimii myös analogisina input-pinneinä (vain 2 pinniä täysin vapaassa käytössä)
- Ohjelmointi tapahtuu USB-johdon tai Bluetooth-yhteyden avulla
- Laitteisto vaatii toimiakseen 3V jännitteen (USB-johdon kautta tai paristolla) ja se antaa ulos maksimissaan 3,3V (USB:n kanssa) tai 3V (pariston kanssa)



- micro:bit:issa on integroituna 3 painonappia, LED-taulu, valosensori, lämpötilasensori, kiintyvyyssanturi, kompassi sekä Bluetooth-radio

STARTER JA INVENTOR KITS

- Koska Arduino UNO:ssa ei ole sisäänrakennettuja input/output-komponentteja, kehitysalustan käyttämiseen tarvitaan jonkinlainen kokoelma lisäkomponentteja kuten ledejä, vastuksia, potentiometreja, valovastuksia, moottoreita, koekytkentälevyjä, johtimia jne.
- Markkinoilla onkin runsaasti erilaisia Arduino Starter/InventorKit:ejä, jotka sisältävät myös itse Arduino UNO:n
- Suurinta osaa Arduino-komponentteja voi käyttää myös micro:bit:in kanssa mutta tarjolla on myös erityisesti micro:bit:ille tarkoitettuja Inventor's Kit:ejä (nämä eivät yleensä sisällä micro:bit-alustaa).
- Komponentit voi kiinnittää micro:bit:in leveisiin liitäntäpinneihin kapeilla hauenleukajohtimilla mutta monimutkaisempien kytkentöjen kanssa on kätevää käyttää erillistä *Breakout board*:ia, jonka avulla myös ohuimmat pinnit saa käyttöönsä. Näitä voi ostaa erillisinä (sisältyy myös kuvassa näkyvään *micro:bit Inventor's Kit*:iin).



Arduino Starterkit:ejä



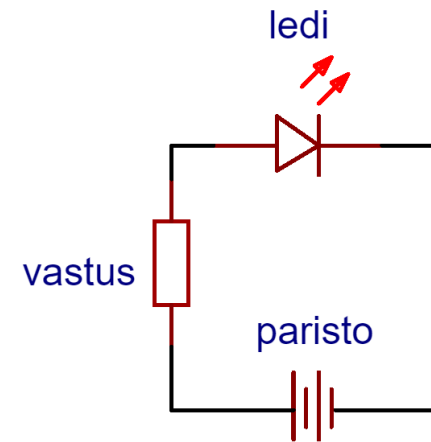
micro:bit Inventor's Kit

LISTA PERUSHARJOITUKSIIN TARVITTAVISTA KOMPONENTEISTA

Komponentti	Kpl	Harj.	Komponentti	Kpl	Harj.
Arduino UNO tai micro:bit + Breakout board	1	Kaikki	Painokytkin	3	5
Koekytkentälevy	1	Kaikki	RGB-ledi	1	5
Johdin (M/M, M/F)	15+10	Kaikki	Asentokytkin	1	8
Vastus (220Ω, 2,2kΩ, 10kΩ)	10+5+5	Kaikki	Ledinäyttö, vain Arduinolle (SparkFun LED Array 8x7)	1	8
Ledi (punainen, keltainen, vihreä)	5+5+5	0, 1, 2, 9	Ultraäänisensori (HC-SR04)	1	7
Valovastus tai valosensori	1	2	IR-vastaanotin ja lähetin, vain Arduinolle	1	9
Lämpötilasensori (TMP36, LM35)	1	3	Transistori (BC 337-25) tai MOSFET (IRF520)	1	8
Pietsosummeri	1	3, 7	Diodi	1	8
Potentiometri (10kΩ)	1	4, 6	DC-moottori	1	6, 10
Kondensaattori (100μF)	1	4	H-silta (L293D)	1	10
Mikroservomoottori	1	4	Paristokotelo 3x1,5V	1-2	6, (7), 10

VIRTAPIIRI

- Virtapiiri on jännitelähteestä (esim. paristosta), johtimista sekä sähkölaitteista koostuva kytkentä, jossa sähkövirta voi kulkea. Sähkövirta aikaansaa laitteesta riippuen valoa, lämpöä, ääntä tai liikettä. Jos virtapiiri on avoin, siinä ei kulje sähkövirtaa.
- Jännitettä (U) mitataan volteissa (V) ja sähkövirtaa (I) ampeereissa (A). Mitä korkeampi jännite on käytössä, sitä suuremmaksi tulee myös sähkövirta. Tätä yhteyttä kutsutaan Ohmin laiksi. Liian suuri sähkövirta kuumentaa johtimia ja saattaa rikkoa herkkiä sähkölaitteita. Siksi on tärkeä osata rajoittaa sähkövirran määrää lisäämällä virtapiiriin vastuksia. Vastukset kuten myös osa muistakin sähkölaitteista vastustavat sähkövirran kulkua. Tätä ominaisuutta kutsutaan resistanssiksi (R) ja sen yksikkö on ohmi (Ω).



Virtapiirejä kuvataan sähköteknisillä piirrosmerkeillä

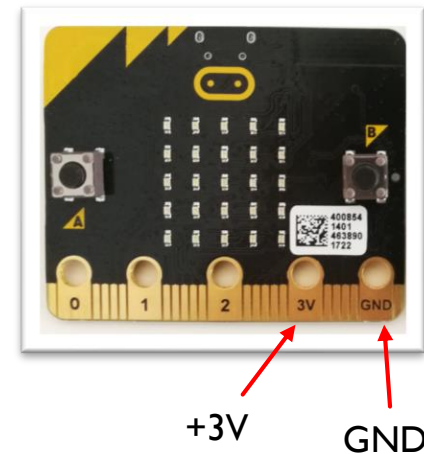
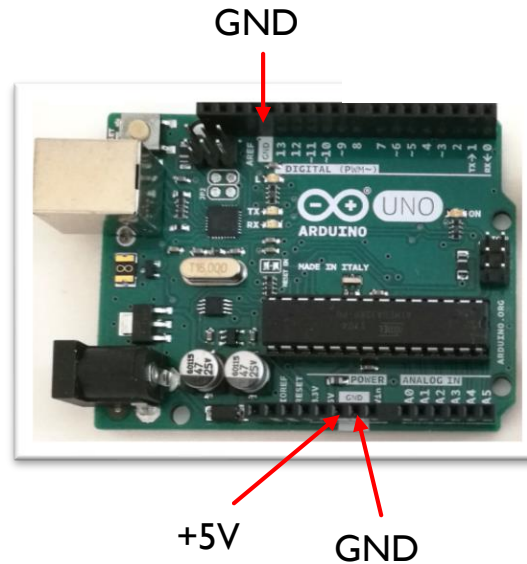
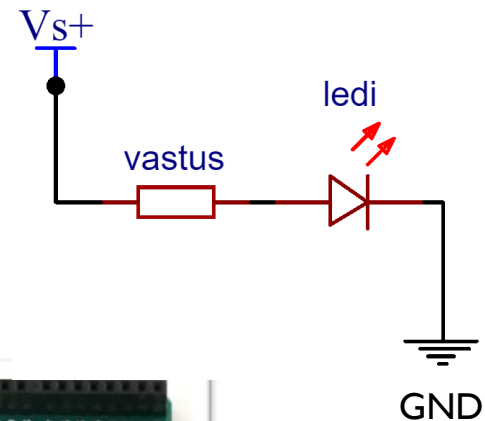
$$R = \frac{U}{I}$$

Ohmin laki

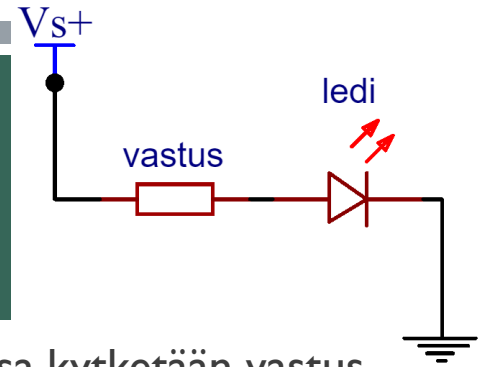
VS+ JA GND

- Sähkövirta on elektronien liikettä. Suljetussa virtapiirissä elektronit virtaavat sähköisen vetovoiman vaikutuksesta pariston negatiiviselta (-) navalta kohti positiivista (+) napaa.
- Ohjelmoitavat kehitysalustat saavat sähköä USB-johdon kautta. Sekä Arduinossa, että micro:bit:issä on lisäksi erilliset pinnit jotka toimivat kuin pariston plus- ja miinusnapa. Arduinossa positiivinen napa on merkitty 5V ja micro:bit:issä 3V ja negatiivisena napana toimii GND (ground). Piirroskuvilla positiivinen napa merkitään V_{s+} merkinnällä ja GND:lle käytetään maadoituksen symbolia.

Suljetun virtapiirin piirroskuva, jossa pariston tilalla on V_{s+} ja GND



VASTUKSEN MITOITTAMINEN



- Sähkövirta saadaan rajoitettua laitteelle sopivaksi, kun sen kanssa kytketään vastus sarjaan (ks. kuva). Silloin vastuksen ja laitteen resistanssit lasketaan yhteen ja virtapiirissä jännitehäviö jakautuu näiden laitteiden kesken.
- Ohmin lain avulla voimme laskea millainen vastus esim. ledillä pitää olla, liian suuri sähkövirta ei käräyttäisi sitä pilalle.
- Jotta mitoittaminen onnistuu, meidän pitää tietää kuinka suuri sähkövirta virtapiirissä saa maksimissaan kulkea. Arduinon I/O pinnistä voidaan turvallisesti ottaa 20mA virta. Jos myös ledi kestää maksimissaan 20mA, voi olla parempi käyttää laskuissa varmuuden vuoksi virralle arvoa 15mA. Jos esim. punaisen ledin jännitehäviö on 2,2V saadaan tarvittava lisävastuksen resistanssi laskettua:

$$R = \frac{U - U_{led}}{I} = \frac{5V - 2,2V}{0,015A} = \frac{2,8V}{0,015A} = 186,66 \dots \Omega$$

- Ledien jännitehäviöt kuitenkin vaihtelevat värin ja valmistusmateriaalin mukaan. Pienimmillään se voi olla 1,7V. Tällä arvolla laskettuna resistanssiksi saadaan 220 Ω , ja tätä arvoa käytämmekin jatkossa kaikissa ledien kytkennöissä.

VASTUKSEN MITOITTAMINEN

- Vastusten arvot pitää mitoittaa micro:bitille erikseen, koska sillä on pienempi lähdejännite (3,3V). Myös sen I/O pinnit kestävät vähemmän virtaa, vain 5mA verran

- Vastuksen mitoitus punaiselle ledille ($V=2,2V$) micro:bitin kanssa olisi:

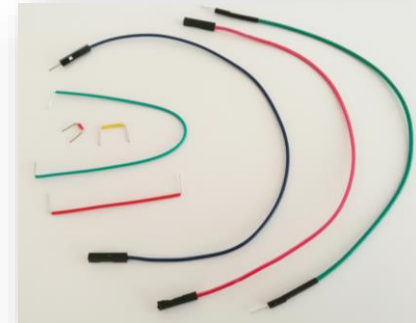
$$R = \frac{U - U_{led}}{I} = \frac{3,3V - 2,2V}{0,005A} = \frac{1,1V}{0,005A} = 220\Omega$$

- HUOM! Jos käytössäsi on micro:bit sekä ledejä, joiden jännitehäviö on alle 2,2V sinun pitää ottaa se huomioon, ja käyttää suurempia vastuksia kuin mitä näissä ohjeissa! Esim. ledin jännitehäviöllä 1,7V lisävastuksen resistanssin arvoksi tulisi 320Ω.

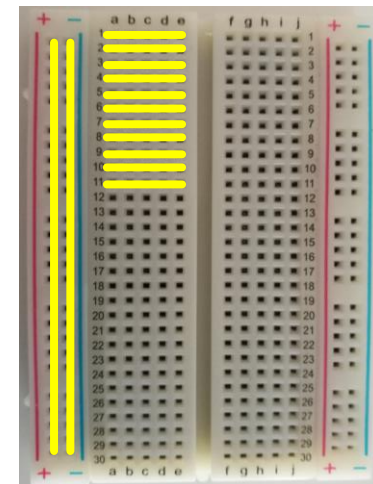
Ledin väri	Jännitehäviö
Punainen	1,7 – 2,4V
Keltainen	1,7 – 2,4V
Vihreä	1,8 – 3,5V
Sininen	2,8 – 3,5V

KOEKYTKENTÄLEVY

- Virtapiiriin voi muodostaa monella tavalla. Johtimina voidaan käyttää hauenleukajohtoja, mutta usein komponentteja on niin paljon, että on käytännöllisempää kiinnittää ne koekytkentälevyyn ja käyttää ohuita johtimia (hyppylankoja).
- Käytettävät komponentit painetaan koekytkentälevyn koloihin. Koekytkentälevyssä vaakarivein reiät a-b-c-d-e ja f-g-h-i-j on yhdistetty toisiinsa metallijohtimilla. Samoin + ja – merkillä merkittyjen pystyraiteiden reiät on yhdistetty toisiinsa. Koekytkentälevyn ansiosta tarvitsemme paljon vähemmän johtimia, ja kytkennästä tulee paljon selkeämpi.
- Pystyraiteeseen vedetään yleensä johdot kehitysalustan V_{s+} :sta (+) ja GND:sta (-).



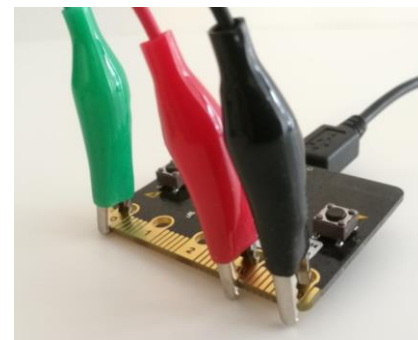
Johtimia (hyppylankoja) koekytkentälevylle



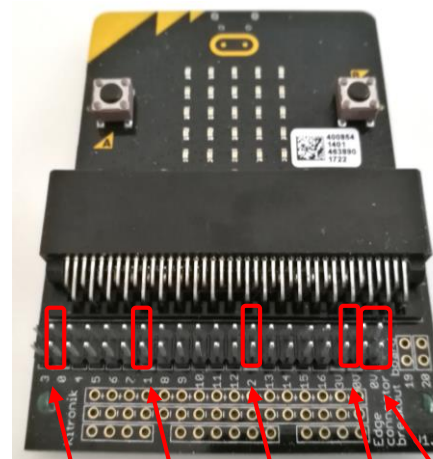
Koekytkentälevyn sisällä kulkee metallijohtimia

BREAKOUT BOARD MICRO:BITILLE

- micro:bitin leveisiin liitäntäpinneihin voi kiinnittää hauenleukajohdot, jos on huolellinen sen kanssa että hauenleuka ei vahingossa osu vieressä oleviin kapeisiin pinneihin (katso kuvasta suositeltu kiinnitystapa).
- Hauenleukajohtoja kätevämpi tapa on ostaa erillinen Breakout board (alempi kuva) johon micro:bit työnnetään. Nyt myös kapeisiin liitäntäpinneihin saa liitettyä johtimen (kuvassa olevat piikit). Huomaa, että Breakout boardin kanssa täytyy käyttää naaras-uros-johtimia (F/M) koekytkenälevyn kanssa.
- Käytännössä täysin vapaasti käytettäviä pinnejä ovat vain: P0, P1, P2, P8 ja P16. Jos nämä eivät riitä, lisäpinnejä saa käyttöönsä, jos luopuu esim. ledinäytön käytöstä tai painonapeista. Tarkemman kuvauksen micro:bit:in pinneistä ja niiden käyttötarkoituksista saat [täältä](#).



Hauenleukojen kiinnitys



P0 P1 P2 +3V GND

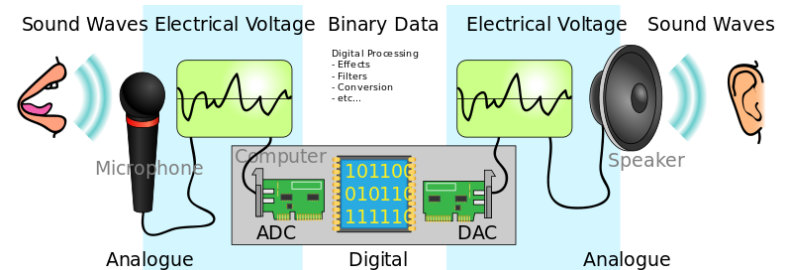
ANALOGISET JA DIGITAALISET PINNIT

- Input ja output laitteet kiinnitetään kehitysalustan GPIO pinneihin (*General Purpose Input/Output pins*). Osa pinneistä toimii sekä input että output pinneinä, osa vain input pinneinä. Input pinneihin kiinnitetään erilaisia sensoreita ja painonappeja, joiden avulla laite saa tietoa ympäristöstä. Output pinneihin kiinnitetään ohjattavia laitteita kuten ledivaloja tai summereita. Jos laitteen ohjaamisessa on vain kaksi vaihtoehtoa: on/off tai sensori-tiedolla on vain kaksi vaihtoehtoa (on/off, I/O), voidaan käyttää digitaalista pinniä. Jos halutaan tutkia esim. lämpötilasensorin antamia arvoja tai ohjata moottorin pyörimisnopeutta tarvitaan pinniä, joka kykenee analogiseen toimintoon. Nyt nollan ja ykkösen sijaan käytössä ovat kokonaisluvut välillä 0-255 (Arduino analogWrite) tai 0-1023 (micro:bit, Arduino analogRead).



Tietokoneet käsittelevät tietoa digitaalisessa muodossa eli ykkösinä ja nollina

Kuva:<https://www.flickr.com/photos/christiaanconlen/20607150556>

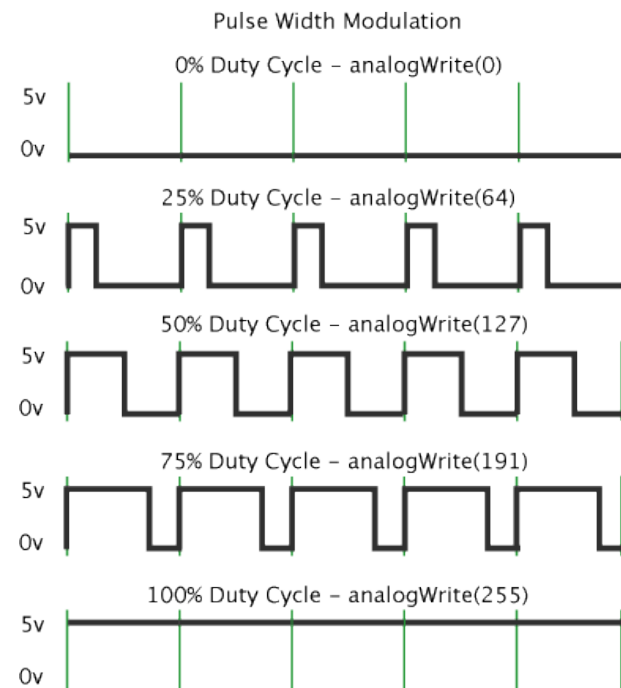


Ääni on analoginen signaali, mutta tietokoneen sisällä sekin on tallennettu digitaaliseen muotoon ykkösiksi ja nolliksi

Kuva:<https://commons.wikimedia.org/wiki/File:CPT-Sound-ADC-DAC.svg>

PULSSINLEVEYSMODULAATIO (PWM)

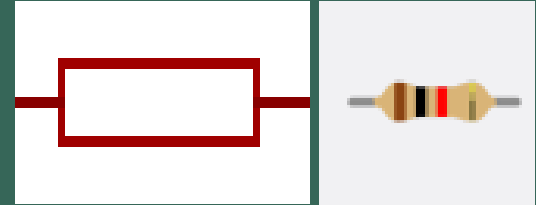
- Kun digitaalinen pinni saa arvon nolla, se vastaa jännitteenä nollaa (GND). Digitaalisen pinnin arvo yksi vastaa ko. laitteen lähdejännitettä (V_s+).
- Kun on kyse analogisista pinneistä, jännite voi saada muitakin arvoja. Kun analoginen pinni saa arvon nolla, se vastaa jännitteenä nollaa (GND) ja analogisen pinnin maksimiarvo 255 (tai 1023) vastaa ko. laitteen lähde-jännitettä (V_s+). Koska kyseessä on digitaalinen laitteisto, analogisen arvon tekeminen pitää tehdä hieman kiertotietä. Tämän tekniikan nimi on pulssin leveysmodulaatio (*Pulse-Width Modulation, PWM*). Siinä pinnin jännite (V_s+) on päällä tietyn aikaa, ja pois tietyn aikaa (GND). Kun näitä päällä/pois jaksoja vaihdellaan hyvin tiheästi, saadaan aikaan jännite, joka on keskimäärin se mitä halutaankin. Pinnejä joissa tämä toiminto on saatavilla merkitään Arduinossa ~merkillä.



PWM-esimerkki arvoilla 0-255 (Arduino)

Kuva:https://commons.wikimedia.org/wiki/File:Pwm_5steps.gif

VASTUKSET



- Vastusten (*resistor*) avulla voimme rajoittaa virtapiirissä kulkevan sähkövirran suuruutta. Elektroniset komponentit kestävät yleensä vain tietyn suuruisen sähkövirran, ja suuremmilla virran arvoilla ne tuhoutuvat. Tuhoutumisen syy on usein ylikuumentuminen, kun liian suuri määrä elektroneja kulkee komponentin läpi liian nopeasti.
- Vastuksen suuruus ilmoitetaan resistanssina (R) ja sen yksikkö on ohmi (Ω). Sen voi myös itse mitata yleismittarilla. Jotta virtapiiriin saadaan sopiva sähkövirran määrä, vastukset pitää mitoittaa oikein. Mitoittaminen tehdään selvittämällä käytetty jännite (U), haluttu sähkövirta (I) ja laskemalla vaadittu resistanssi (R) käyttämällä Ohmin lakia: $R=U/I$. Kytkennän kokonaisvastuksen suuruuteen vaikuttaa myös se ovatko komponentit kytketty sarjaan vain rinnan.
- Muista: älä koskaan kytke V_s+ :aa suoraan GND:hen ilman vastusta (aiheutuu oikosulku ja laite voi tuhoutua)!



Vastuksia



Resistanssin mittaaminen yleismittarilla

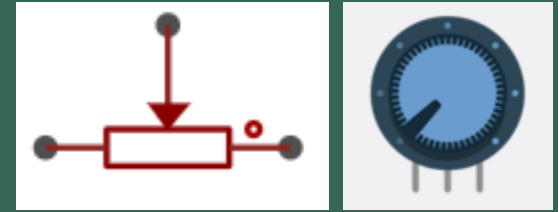
VASTUKSEN RESISTANSSIN LUKEMINEN

- Kytkenäkaavioissa kerrotaan tarvittavan vastuksen suuruus. Huomaa, että merkintä 220R tarkoittaa samaa kuin 220Ω , samoin merkintä 10k tarkoittaa $10k\Omega$. R kirjainta käytetään toisinaan myös pilkun tilalla joten 4R7 tarkoittaa $4,7\Omega$ ja 4k7 tarkoittaa $4,7k\Omega$.
- Vastuksen resistanssi on merkitty siihen värillisten viivojen avulla. Neljän viivan järjestelmässä resistanssi luetaan kolmesta ensimmäisestä viivasta ja viiden viivan järjestelmässä neljästä ensimmäisestä viivasta (tarkempi). Näistä viimeinen viiva määrittää kertoimen. Reunimmaisena oikealla oleva viiva kertoo toleranssin. Resistanssin selvittämiseen on tarjolla myös kätevä [nettityökalu](#).

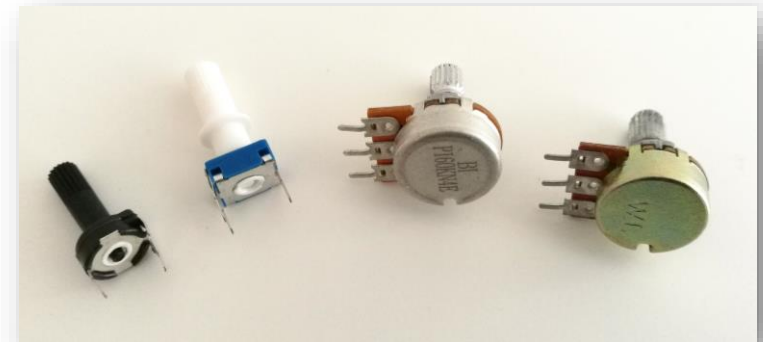
Resistanssi	Neljä viivaa	Viisi viivaa
220 Ω		
4,7k Ω		
10k Ω		

	Luku	Kerroin
	0	1 Ω
	1	10 Ω
	2	100 Ω
	3	1k Ω
	4	10k Ω
	5	100k Ω
	6	1M Ω
	7	10M Ω
	8	100M Ω
	9	1G Ω
	-	0,1 Ω
	-	0,01 Ω

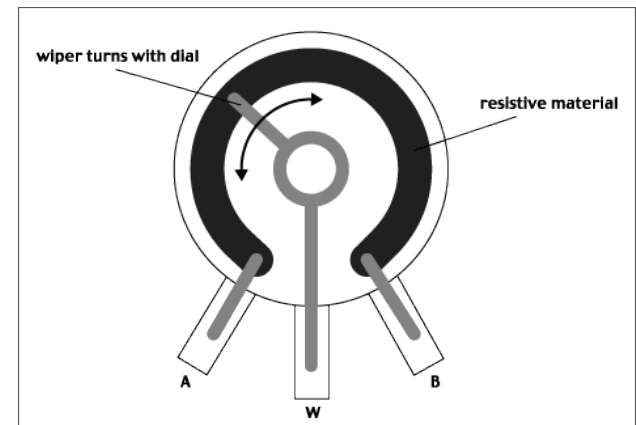
POTENTIOMETRI



- Potentiometri (*potentiometer*) on mekaaninen säätövastus
- Vastuksen resistanssia voidaan säätää portaattomasti kiertämällä nupista
- Potentiometrejä on erilaisia mutta rakennussarjoissa ne ovat yleensä lineaarisia yksikerrospotentiometrejä. Lineaarinen tarkoittaa sitä, että niiden jännite on suoraan verrannollinen nupin asentoon. Tällaisen potentiometrin sisällä on hevosenkengän muotoinen vastus sekä säätönupin mukana kiertyvä viisari, joka jakaa vastuksen kahdeksi eri suuruiseksi vastukseksi.
- Keskimmäinen kohtio (W) kytketään analogiseen IN pinniin, yksi kohtio (A tai B) kytketään maahan (GND) ja toinen lähdejännitteeseen (V_s^+). Yleensä kytkemissuunta on niin että maksimiarvo saadaan kun ”potikka on kaakossa”.

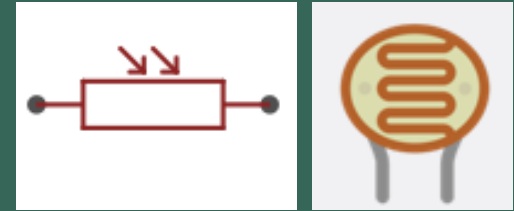


Erilaisia 10kΩ potentiometrejä

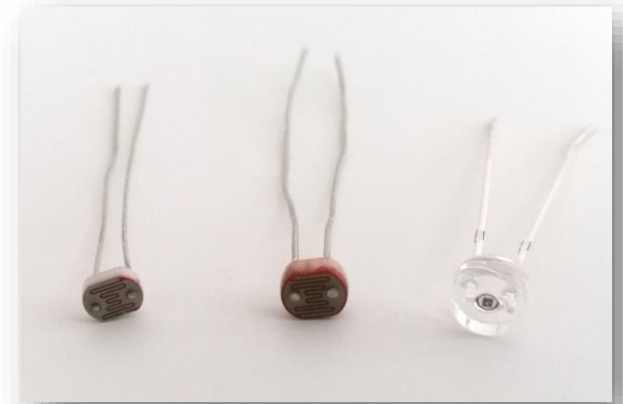


Kuva: <https://i.stack.imgur.com/iiTjO.gif>

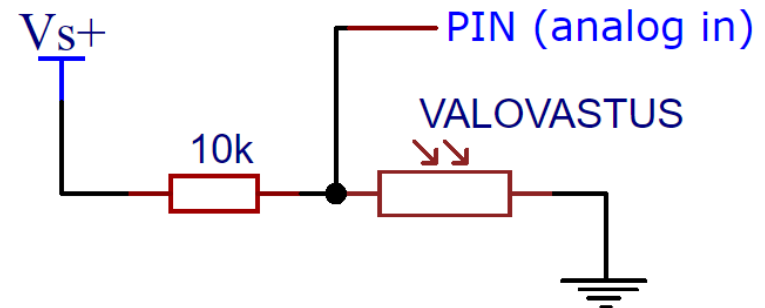
VALOVASTUS



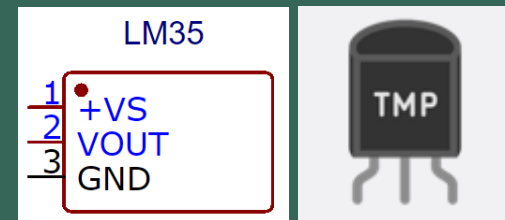
- Valovastus (*photoresistor*) on vastus, jonka resistanssi riippuu siihen kohdistuvan valaistuksen voimakkuudesta
- Valovastuksen resistanssi pienenee, kun valon määrä kasvaa. Tämä johtuu siitä että käytetyn puolijohdemateriaalin sidoselektronit virittyvät korkeampaan energiatilaan, irtoavat kemiallisista sidoksista ja siirtyvät vapaiksi elektroneiksi, jotka pystyvät kuljettamaan sähköä.
- Valovastuksen resistanssi voi vaihdella $1\text{M}\Omega$:sta (pimeää) 100Ω :iin (kirkas valo).
- Valovastus kytketään yleensä sarjaan $10\text{k}\Omega$ vastuksen kanssa. Jännite jakautuu Ohmin lain mukaisesti näiden kahden vastuksen välille ja muuttuva jännite voidaan mitata näiden väliltä (kuvassa: analog in).
- Jossakin rakennussarjoissa valovastuksen tilalla käytetään valosensoria (*ambient light sensor*).



Kaksi valovastusta ja valosensori



LÄMPÖTILASENSORI

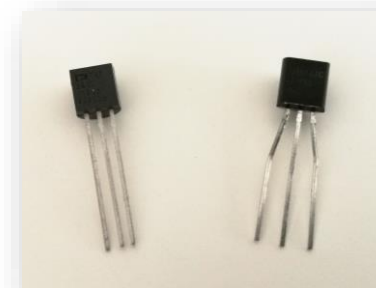


- Lämpötilasensori (*temperature sensor*) on integroitu piiri (IC) eli sen sisällä on useampia komponentteja, jotka käsittelevät signaalia. Sen toiminta perustuu siihen, että muuttuva lämpötila saa sensorin sisällä olevan puolijohdekomponentin käyttäytymään eri tavalla, mikä havaitaan muuttuneena jännitearvona, jota vahvistetaan ja käsitellään niin että ulostuleva jännite (V_{out}) seuraa lineaarisesti lämpötilaa (kuvan lämpötilasensoreilla $10\text{mV}/^{\circ}\text{C}$).
- Jotta sensorin antamasta jännitteestä (x) saadaan lämpötila (T) pitää se syöttää muunnoskaavaan, joka riippuu sensorista. Lisäksi GPIO-pinni ei anna sensorin jännitearvoa suoraan, vaan se skaalaa sen välille 0-1023 käyttöjännitteen ($+V_s$) suhteen. Esim. kun $+V_s$ on 5,0V muunnoskaavat esimerkkisensoreille:

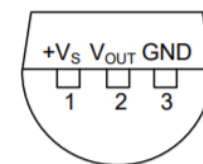
$$T_{LM35}(x) = \frac{5.0 * x * 100.0}{1024}$$

$$T_{TMP36}(x) = \left(\frac{5.0 * x}{1024} - 0.5 \right) * 100.0$$

Huom! TMP36gz sensorin kaavassa on huomioitu se, että sen antamassa lukemassa on 0.5V offset.

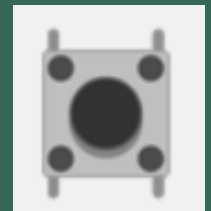
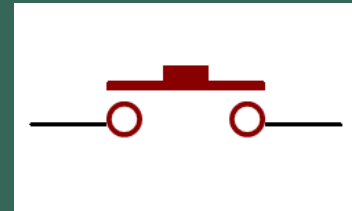


TMP36gz ja LM35



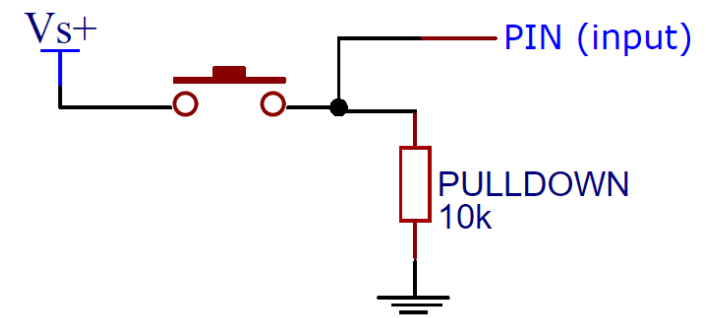
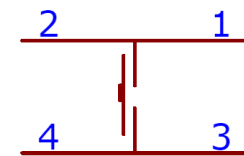
TMP36gz ja LM35 pinnit katsottuna alapuolelta

PAINOKYTKIN

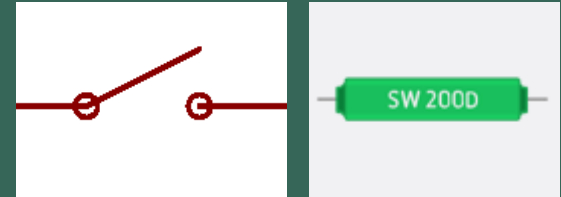


Painokytkimiä

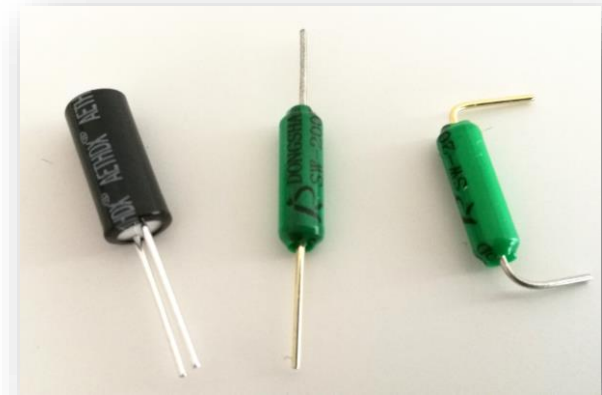
- Painokytkin tai painonappi (*push button/switch*) on yksinkertainen kytkin, joka painettaessa sulkee virtapiirin.
- Rakennussarjojen mukana tulevissa painokytkimissä on yleensä neljä jalkaa, joista jalat 1 ja 2 sekä 3 ja 4 ovat kiinni toisissaan (ks. keskimäinen kuva oikealla).
- Painonappi voidaan kytkeä esim. niin että jalka 2 kytketään $+V_s$ jännitteeseen ($+5V$) ja jalka 3 digitaaliseen INPUT pinniin. Kun kytkin on suljettuna, molemmat jalat ovat samassa jännitteessä $+5V$. Kun kytkin aukeaa, jännite ei kuitenkaan aina laske nolleen vaan jää heittelehtimään. Siksi jalka 3 kytketään $10k\Omega$:n vastuksen kautta maahan (GND). Tätä vastusta kutsutaan PULL DOWN vastukseksi.



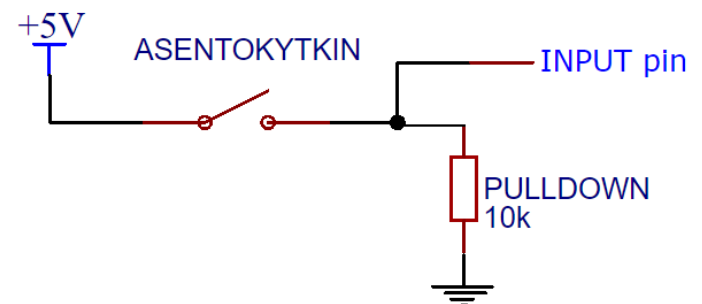
ASENTOKYTKIN



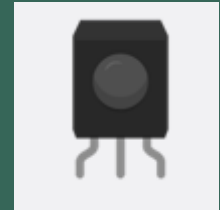
- Asentokytin (*tilt switch*) on yksinkertaisesti sylinteri, jonka sisällä pyörii sähköä johtavasta materiaalista tehty kuula. Sylinterin toisessa päässä on kaksi johdinta ja kun kuula viehän tähän päähän, se sulkee virtapiirin. Kun kuula vierii sylinterin toiseen päähän, virtapiiri avautuu.
- Jos laitteen asennon muutos halutaan havaita x- ja y-akselien suunnassa, tarvitaan kaksi 90-asteen kulmaan kiinnitettyä asentokytintä.
- Asentokytin asennetaan samalla tavalla kuin painokytin eli se kytketään maahan (GND) PULL DOWN vastuksen kautta.



Asentokytimiä



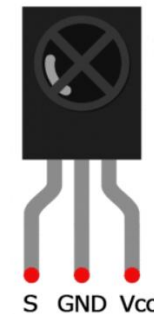
IR-VASTAANOTIN (JA LÄHETIN)



- IR-vastaanotin (*IR-receiver*) vastaanottaa infrapunasäteilyä, joita IR-lähetin (*IR-transmitter*) lähettää. Lähetettävä viesti enkoodataan infrapunasäilylpulsseiksi, jotka vastaanottava pää dekodaa takaisin ymmärrettävään muotoon.
- IR-lähetin on yleensä ledi, joka lähettää silmälle näkymätöntä infrapunavaloa (aallonpituus 940nm). IR-lähettimen ja -vastaanottimen täytyy olla näköetäisyydellä toisistaan.
- IR-vastaanotin on yleensä valotransistori (tavallinen transistori, jossa valo pääsee absorboitumaan kannan ja emitterin väliseen P-N-liitokseen). Tämä aikaansaa elektroni-aukko-parin ja transistori alkaa johtamaan sähköä kollektorilta emitterille.
- Huomaa, että IR-vastaanotin on erittäin herkkä pienillekin sähköhäiriöille, joten sen kanssa tulee helposti ongelmia esim. DC-moottoreiden kanssa.

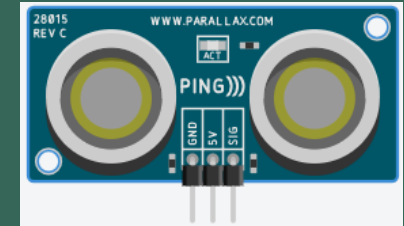
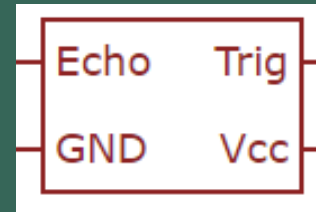


IR-lähettimiä ja -vastaanottimia



IR-vastaanottimen kytkentä

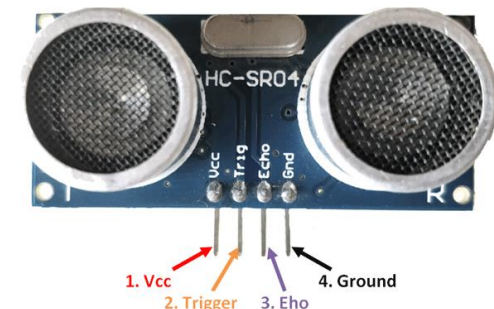
ULTRAÄÄNISENSORI



- Ultraäänisensori (*ultrasonic sensor*) mittaa esineiden etäisyyttä lähettämällä ultraääni-impulsseja tasaisin väliajoin. Nämä impulssit heijastuvat (*echo*) esineistä takaisin. Mittaamalla aika joka kuluu kaiun saapumiseen sensorille, voidaan laskea esineen etäisyys sensorista, kun äänen kulkunopeus ilmassa tunnetaan.
- Sensoria ohjataan Trig- ja Echo-pinnien kautta. Lisäksi sensorin Vcc-pinni kytketään lähdejännitteeseen (V_s) ja GND pinni maahan. Sen toimintasäde on 2-80cm.
- Joissakin ultraäänisensoreissa on vain kolme pinniä: Vcc, GND ja SIG.
- Huomaa, että tämä sensori vaatii toimiakseen 5V jännitteen.

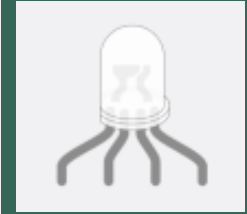
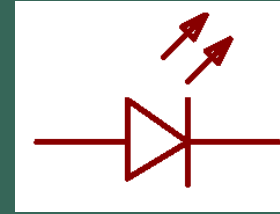


Ultraääni-sensori



Kuva: <https://components101.com/ultrasonic-sensor-working-pinout-datasheet>

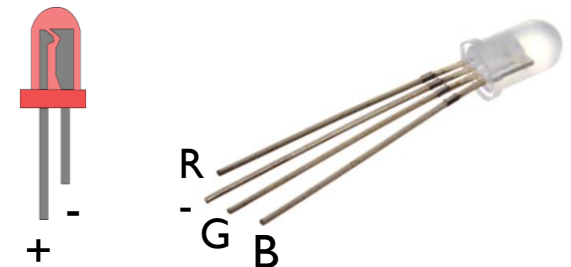
LED JA RGB-LED



- LED tai ledi (*Light-Emitting Diode*) on puolijohdekomponentti, jossa sähkövirta aikaansaa elektroluminesenssireaktion, jossa syntyy valoa.
- LED on muodostettu liittämällä yhteen kahta erityyppistä puolijohdemateriaalia. N-tyypin puolijohteessa on elektronivaje ja P-tyypin puolijohteessa ylimääräisiä elektroneja. Sähkövirta saa ylimääräiset elektronit liikkumaan aineiden rajapinnalle, jossa ne täyttävät N-tyypin puolijohteen elektroniaukkoja. Koska ne ovat alemmalla energiatilalla, elektronien ylimääräinen energia vapautuu fotoneina eli valona. Ledin lähettämän valon aallonpituus/väri riippuu käytetyistä aineista.
- LED, kuten muutkin diodit, johtaa sähköä vain yhteen suuntaan. LED vaatii tietyn kynnysjännitteen toimiakseen (1,7 - 3,7V) ja sen läpi kulkee 20mA. RGB-ledissä on kolme lediä samassa paketissa (Red-Green-Blue).



Erivärisiä ledejä sekä RGB-ledi



Ledin pisin jalka on + mutta RGB-ledillä pisin jalka voi olla ledien yhteinen + tai yhteinen -

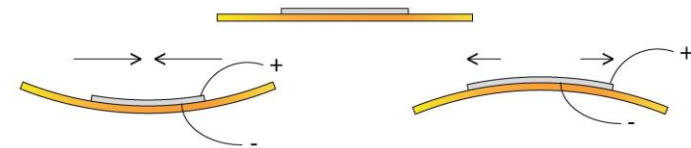
PIETSOSUMMERI



- Pietsosummeri (*Piezo Buzzer*) on komponentti, joka muuntaa sähkökentän muutokset ääneksi. Pietsosummerin sisällä on pietsosähköinen elementti, joka venyy ja kutistuu sähkökentän muutosten mukana. Tämä elementti on kiinni metallilevyssä, joka alkaa värähdellä (alemman kuvan keltainen osa) pietsoelementin tahdissa (harmaa osa) ja tuottaa ääntä sille ominaisella taajuudella.
- Pietsoelementti voi toimia myös toiseen suuntaan, jolloin se reagoi ääneen muodostamalla muuttuvan jännitteen. Tätä ilmiötä kutsutaan pietsosähköiseksi ilmiöksi. Siinä mekaaninen rasitus saa aineen polarisoitumaan sähköisesti (varaukset jakautuvat ja kiteen reunojen välille muodostuu jännite).



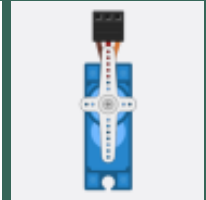
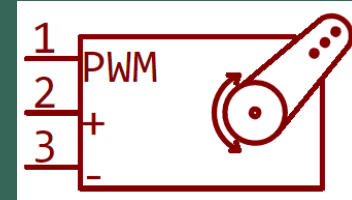
Pietsosummereita



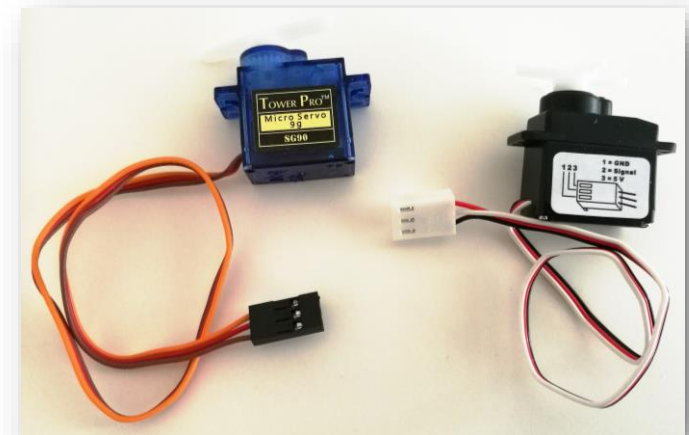
Pietsosummerin toiminta

Kuva:https://commons.wikimedia.org/wiki/File:Piezo_bending_principle.jpg

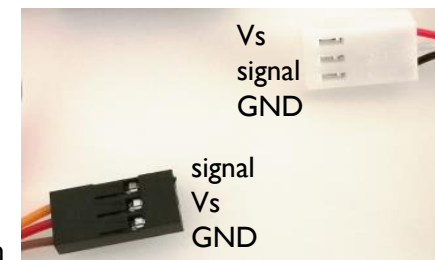
SERVOMOOTTORI



- Servomoottori tai servo (*servomotor*) on moottori, jota ohjataan kääntymään tiettyyn asentoon. Niiden avulla voidaan esim. kääntää kaukoputki osoittamaan oikeaan kohtaan taivaalla.
- Servomoottorissa on elektroninen piiri, joka mittaa moottorin asentokulmaa ja vertaa sitä ohjaussignaalin antamaan kulmaan. Servo kääntyy niin kauan kunnes haluttu asema on saavutettu. Jotta moottorille saadaan analoginen ohjaussignaali digitaalisen portin avulla, tarvitaan pulssinleveysmodulaatiota (PWM). Tämä toiminto löytyy Arduinoista porteista joissa on ~ merkinä ja micro:bitin porteista P0, P1, P2 ja P4.
- Rakennussarjojen servomoottoreita voidaan ohjata kääntymään 0-180° ja ne vaativat yleensä 4,8-6,0V jännitteen.



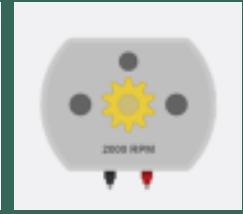
Mikroservomoottoreita



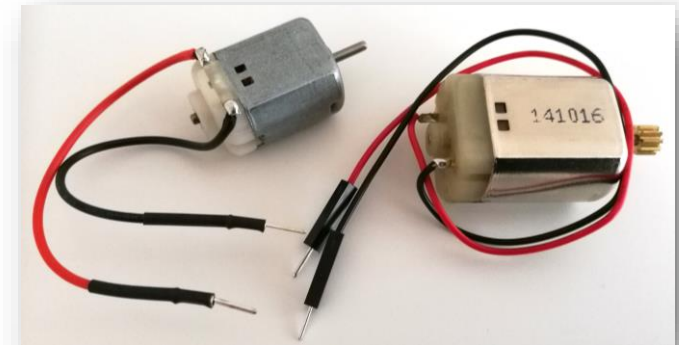
oranssi
punainen
ruskea

punainen
valkoinen
musta

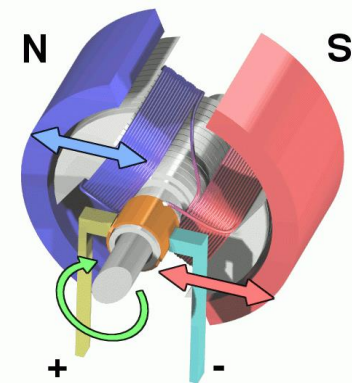
DC-MOOTTORI



- DC-moottori eli tasavirtamoottori (*DC motor*) muuntaa sähköenergiaa liike-energiaksi. Sen toiminta perustuu sähkömagneettien avulla saataviin veto- ja hylkimisvoimiin. Sähkömoottorin pyörimissuunta riippuu sähkövirran kulkusuunnasta.
- Tasavirtamoottori pyörivä osa, roottori, toimii rautasydämenä ja sen ympärille kierretään kuparijohtoa käämiksi. Kun kuparijohdossa kulkee virta, roottori muuttuu sähkömagneetiksi. Moottorin paikallaan pysyvä osa eli staattori on joko kestopagneetti tai sähkömagneetti. Jotta roottori saadaan pyörimään, virran suunnan ja samalla sähkömagneetin veto- ja hylkimisvoimien suuntien on muututtava pyörimisen aikana. Tämä saadaan aikaan kommutaattorilla eli mekaanisella virran kääntäjällä (grafiittiharjoja).

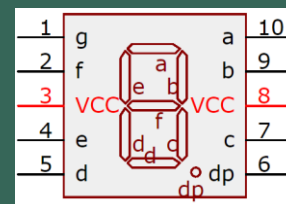
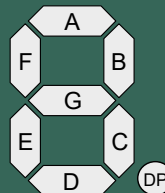


Tasavirtamoottoreita



Kuva: https://upload.wikimedia.org/wikipedia/commons/8/89/Electric_motor.gif

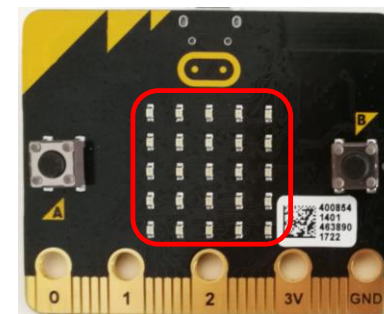
LED-NÄYTTÖ



- LED-numeronäytössä (*eight-segment display*) on 8 lediä, jotka voidaan kytkeä päälle ja pois toisistaan riippumatta. Riippuen siitä mitkä ledit ovat päällä näytössä näkyy numero tai kirjain (A, b, c, d, E, F). ledeillä on joko yhteinen anodi (V_s^+) tai katodi (GND).
- LED-matriisissa ledit ovat vaaka- ja pystyriveissä, joten sen avulla voidaan esittää monimutkaisempia kuvioita. LED-matriiseissa ledejä ei ohjata yhtä kerrallaan (koska OUTPUT-pinnit eivät riitä) vaan niiden ohjaaminen multipleksataan. Tämä tarkoittaa sitä, että ledejä ohjataan yhtä kerrallaan peräkkäin, eli niitä sytytellään ja sammutellaan vuorotellen niin nopeassa tahdissa, että syntyy illuusio joidenkin ledien palamisesta. Tämä on jo sen verran monimutkaista, että ohjaamiseen tarvitaan usein oma kirjasto.

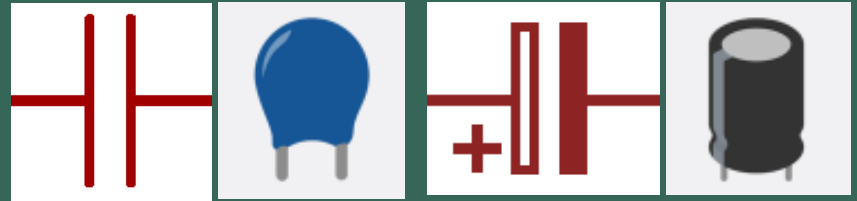


LED-numeronäyttöjä sekä 8x8 ja 8x7 LED-matriisit

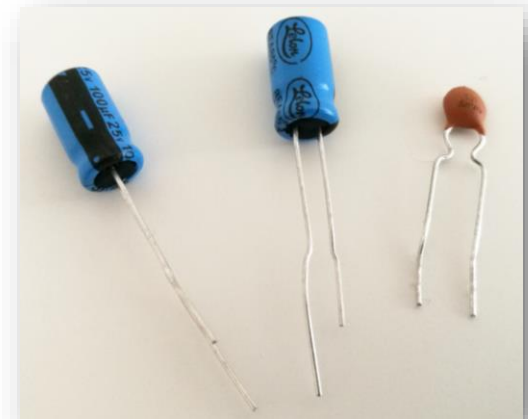


micro:bit'in 5x5 LED-matriisi

KONDENSAATTORI

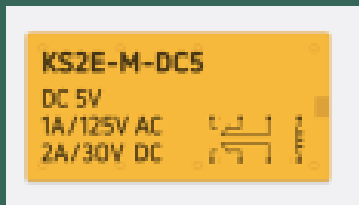
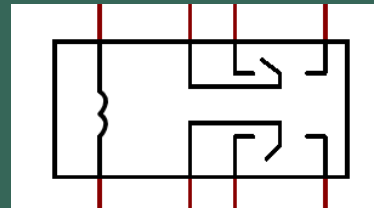


- Kondensaattorit (*capasitors*) ovat komponentteja, joita käytetään mm. jännitevaihteluiden tasaamiseen. Ne varastoivat sähköenergiaa sähkökenttään, joka muodostuu kahden elektrodin välille (levyjä tai pallokuoria). Kun kondensaattori kytketään jännitelähteeseen, sen elektrodit saavat saman suuruisen mutta eri merkkisen varauksen. Kun kondensaattori kytketään kuormaan, sähkövaraus purkautuu sähkövirtana.
- Kondensaattoreita on erilaisia. Elektrolyyttikondensaattorissa (valokuvassa vasemmalla) levyjen väliin muodostuu metallioksidikerros eristeeksi. Jos kondensaattorin navat kytkee väärinpäin se voi tuhoutua, koska oksidikerrosta ei synny kunnolla ja levyt menevät oikosulkuun. Keraamisilla kondensaattoreilla kytkentäsuunnalla ei ole väliä.
- Varaus säilyy kondensaattorissa pitkään, joten siitä voi saada sähköiskun, jos varausta ei pureta oikein.



Elektrolyyttikondensaattoreita ja keraaminen kondensaattori

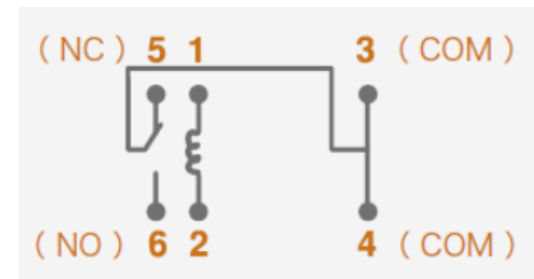
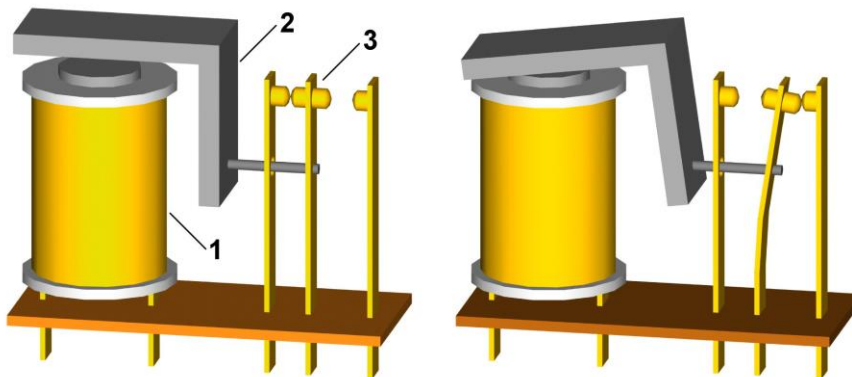
RELE



- Rele (*relay*) on sähköisesti ohjattavissa oleva sähkömekaaninen kytkin. Sen avulla voidaan ohjata suuremman jännitteen piirejä (yleensä AC) pienemmän jännitteen ohjausvirralla (DC). Releen sisällä on käämi (1). Kun ohjausvirta kulkee käämin läpi se muuttuu sähkömagneetiksi, joka vetää releen liikkuvaa rautakielekettä (2) puoleensa ja avaa tai sulkee virtaviirin riippuen siitä miten rele on kytketty.



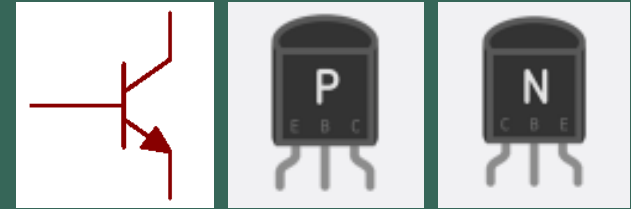
Rele



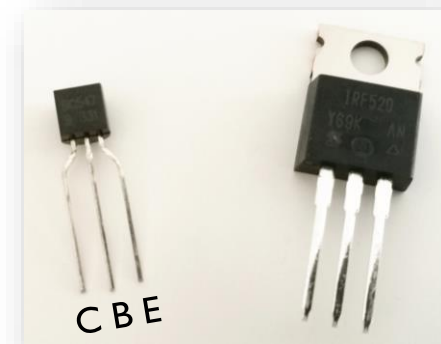
Kuusijalkaisen releen kytkentä
(3 ja 4 ovat samat)

Kuva: https://commons.wikimedia.org/wiki/File:Relay_principle_horizontal.jpg

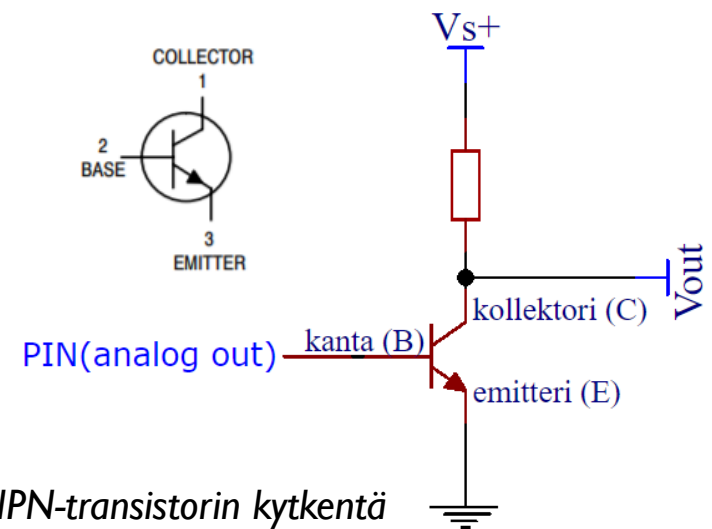
TRANSISTORI



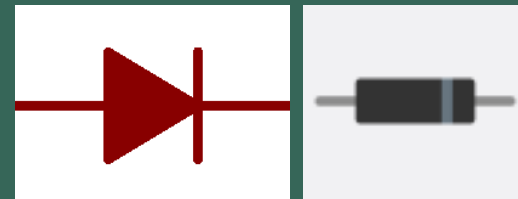
- Transistori (*transistor*) on puolijohdekomponentti, joka voi toimia kytkimenä tai vahvistimena. Siinä on kolme liitintä: kanta (B), kollektori (C) ja emitteri (E). Kantaan kytketään pieni ohjausvirta, jonka avulla ohjataan sähkövirran suuruutta kollektorin ja emitterin välillä.
- Transistoreita on monenlaisia. NPN- ja PNP-transistoreissa on liitetty yhteen P- ja N-tyypin puolijohteita. Erona näissä on virran suunta ja jännitteen napaisuus. NPN-transistorissa emitteriltä kulkee elektroneja kohti kollektoria, PNP-transistorissa puolestaan aukkoja. Näiden lisäksi on monimutkaisempia FET-transistoreja (mm. MOSFET).
- Transistoreita käytetään mm. vahvistimissa sekä moottoreiden ohjauksessa.



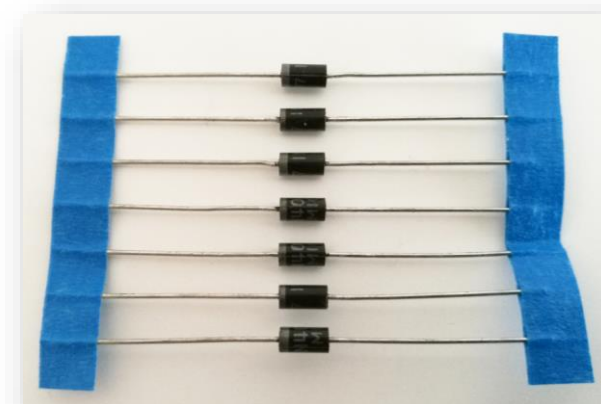
NPN-transistori sekä MOSFET



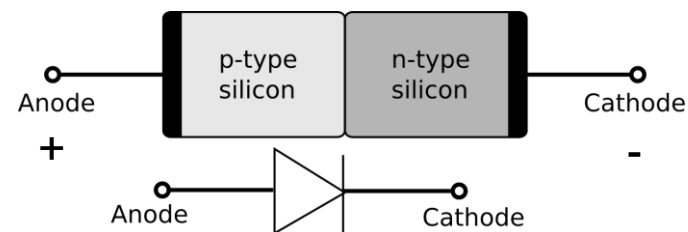
DIODI



- Diodi (*diode*) on puolijohdekomponentti, joka päästää sähkövirran kulkemaan lävitseen vain toiseen suuntaan. Se koostuu N- ja P-tyypin puolijohteista, jotka on liitetty yhteen. Kun diodi on kytketty estosuuntaan, sen läpi ei kulje virtaa, koska sähkövarausten kuljettajat vetäytyvät pois N-P-rajapinnalta sähköisten vetovoimien takia. Päästösuuntaan kytketyn diodin läpi kulkee virta kun jännite nousee yli sen kynnysjännitteen ($\sim 0,7V$).
- Diodeja käytetään mm. vaihtovirran tasasuuntaamiseen sekä moottoreiden ohjauksessa estämään moottorin virran syötön loputtua syntyvä induktiovirta, joka voi rikkoa muita komponentteja kuten transistoreja.
- Ledit ovat myös diodeja (ks. LED ja RGB-LED).



Diodeja

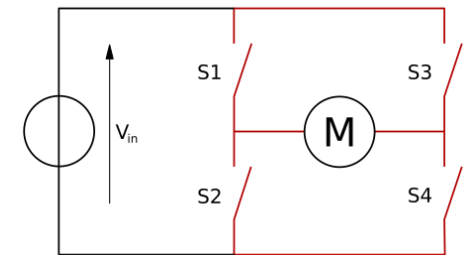


Kuva:https://bg.m.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:PN_diode_with_electrical_symbol.svg

H-SILTA

- H-silta on integroitu piiri (IC), jota käytetään DC-moottorin ohjaamiseen. Sen avulla moottori saadaan pyörimään eri nopeuksilla sekä vaihtamaan pyörimissuuntaansa. H-silta koostuu neljästä kytkimestä, jotka saadaan aikaan käyttämällä NPN/PNP-transistoreita tai MOSFET-transistoreita. Kun kytkimiä suljetaan pareittain (S1+S4 tai S2+S3), virta saadaan kulkemaan eri suuntaan moottorin läpi. Huomaa, että kytkimet S1+S2 ja S3+S4 eivät saa olla samaan aikaan kiinni (oikosulku!). Piirissä on lisäksi käytetty diodeja suojelemaan transistoreja moottorin virransyötön loputtua synnyttämältä induktiovirralla (moottorista tulee generaattori) sekä kondensaattori tasaamaan moottorin käynnistyksen aiheuttamia virtapiikkejä. H-sillan transistorien avulla pienen ohjausvirran avulla voidaan ohjata DC-moottorin vaatimaa suurempaa sähkövirtaa.

H-sillan nimi tulee kytkennän muodostamasta H:sta, virta kulkee kun S1 ja S4 suljetaan, ja sen suunta muuttuu kun S3 ja S2 suljetaan (muut avataan)

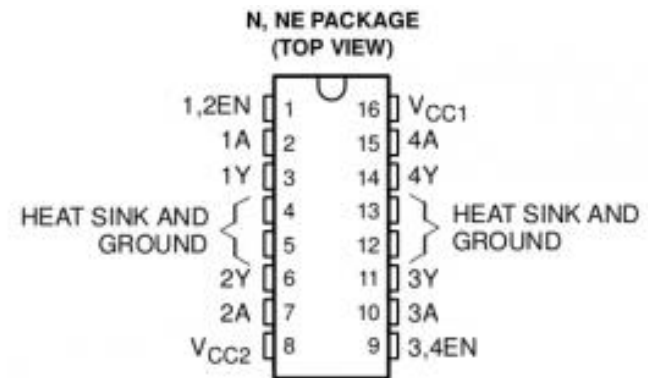


Kuva: https://upload.wikimedia.org/wikipedia/commons/d/d4/H_bridge.svg

H-SILTA L293D



- V_{cc1} (pinni 16) kytketään Arduinon/pariston +5V (tämä tarvitaan H-sillan omaan toimintaan)
- V_{cc2} (pinni 8) kytketään ulkoiseen jännitelähteeseen, yleensä paristoon (maksimissaan 36V)
- 1,2 EN (pinni 1) ja 3,4 EN (pinni 9) ovat moottoreiden nopeuspinnejä (*enable pins*). Nämä kytketään PWM pinneihin (analog out), jotka on merkitty Arduinossa ~, microbitissä pinnit P0-P4
- 1A (pinni 2) ja 2A (pinni 7) ovat moottorin pyörimissuunnan ohjauspinnejä moottorille 1. Nämä kytketään Arduinoon.
- 1Y (pin 3) ja 2Y (pinni 6) ovat moottorin ulostulopinnot ja ne kytketään moottoriin 1.
- 3A (pinni 10) and 4A (pinni 15) ovat moottorin pyörimissuunnan ohjauspinnejä moottorille 2. Nämä kytketään Arduinoon.
- 3Y (pinni 11) and 4Y (pinni 14) ovat ulostulopinnejä moottorille 2. Nämä kytketään moottoriin 2.
- Pinnot 4,5,12,13 kytketään maahan (GND)



Arduino Starter Kitin H-sillalla (L293D) voidaan ohjata kahta moottoria pyörimään eteen- tai taaksepäin eri nopeuksilla (ks. [Tutoriaali 1](#) ja [Tutoriaali 2](#)).

MOOTTORIN OHJAAMINEN H-SILLALLA

■ Jokaista moottoria ohjataan siis kolmella pinnillä: 1 nopeuspinni (EN pinni) ja kaksi ohjauspinniä

■ Jos nopeuspinnin (EN pinni) arvo on nolasta poikkeava, moottori pyörii

■ Jos ohjauspinni 1 on HIGH ja ohjauspinni 2 on LOW, niin moottori pyörii yhteen suuntaan

■ Jos ohjauspinni 1 on LOW ja ohjauspinni 2 on HIGH, niin moottori pyörii toiseen suuntaan

■ Muissa tapauksissa moottori ei pyöri ollenkaan

■ Moottori pyörii sitä nopeammin mitä suurempi nopeuspinniin (EN pinni) kirjoitettu arvo on

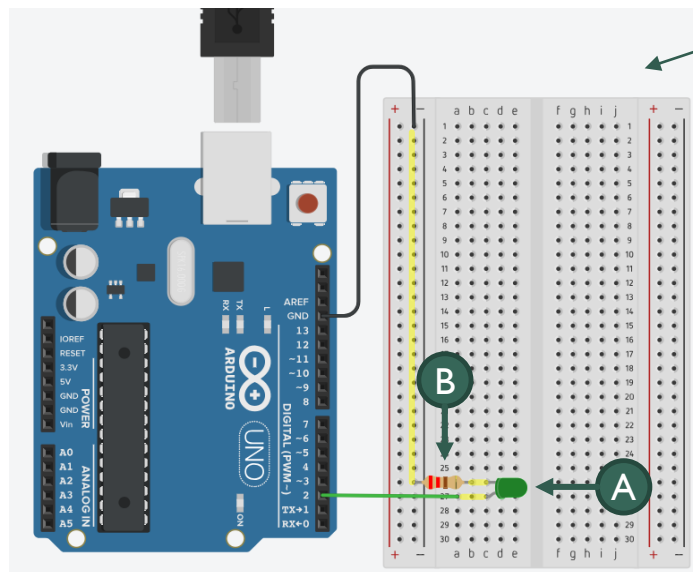
EN PINNI	PINNI 1	PINNI 2	MOOTTORI
1-255	1	0	Pyörii yhteen suuntaan
1-255	0	1	Pyörii toiseen suuntaan
1-255	1	1	Ei pyöri
0	-	-	Ei pyöri

TYÖKORTIT

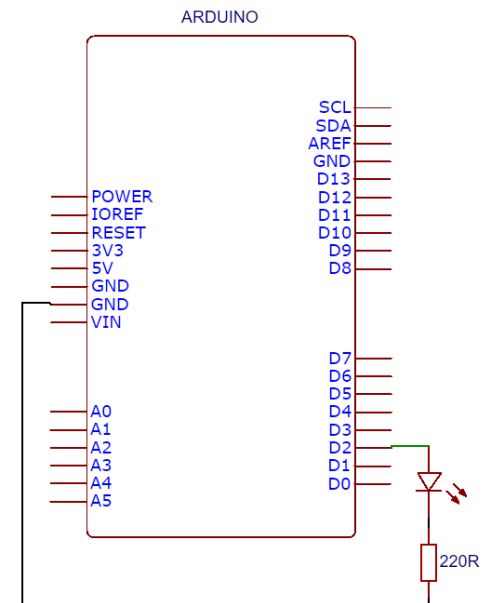
Työ	Komponentit	Ohjelmointi
0. Ledin vilkuttelua	Ledi	Valmiin koodin käyttäminen
1. Ledit liikennevaloina	Kolme lediä	Vakio (muuttuja), silmukka, int, digital out
2. Automaattinen ledimajakka	Valovastus	Muuttuja, ehtolause, double, analog in, sarjamonitori
3. Lämpötilavahti	Lämpötilasensori, pietso-summeri	Funktio, analog out
4. Potentiometrillä ohjattu servo	Potentiometri, kondensaattori, servomoottori	Kirjaston käyttäminen
5. Värivalojen ohjauspaneeli	Painonapit, RGB-ledi	Boolean operaattorit (and, or)
6. Potentiometrillä ohjattu DC-moottori	Transistori/MOSFET, diodi, DC-moottori, potentiometri	
7. Liikeohjattava soitin	Ultraäänisensori, pietso-summeri	Taulukko, kirjaston asentaminen
8. Asennon tunnistin	Asentokytkin, ledinäyttö	Byte, const, static,
9. Kauko-ohjatut ledivalot (Arduino)	IR-vastaanotin ja -lähetin, ledejä	Totuusarvot (true/false)
10. DC-moottorin ohjaus H-sillan avulla	H-silta, DC-moottori	For – silmukka, define

0. LEDIN VILKUTTELUA KYTKENNÄT

- Tässä harjoituksessa kytket ledin vastuksen kanssa ohjelmitavaan laitteeseen ja lataat ledin_vilkuttelua -ohjelman siihen valitsemallasi ohjelmointiympäristöllä.
- Tee kytkentä oheisen mallin mukaan. Jos et ole varma miten johtimet on kytkettävä, tarkista asia sähköteknisestä piirroksesta.
- Tarkemmat tiedot käytettävistä komponenteista ja pinneistä löydät oheisesta taulukosta.



Tähän kytkentäkuvaan on lisätty keltaisia viivoja, jotta syntyvä virtapiiri olisi helpompi hahmottaa (kytkentälevyn sisäisiä johteita).

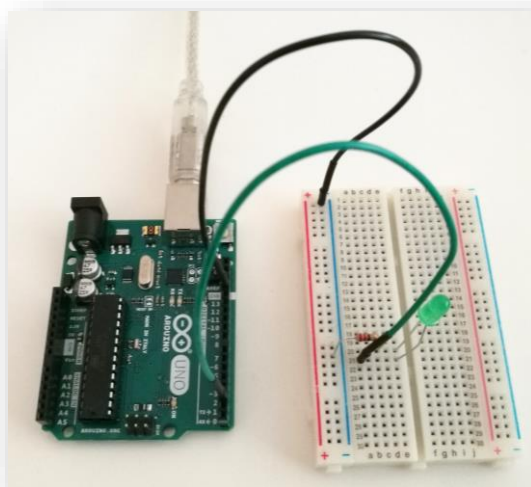


Komponenttilista ja pinnien numerot

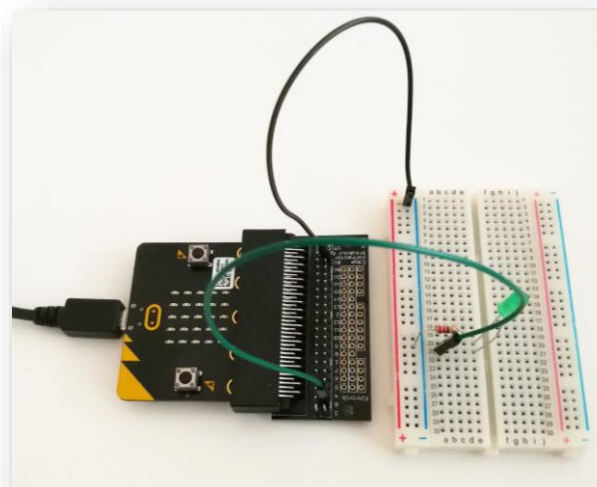
		Arduino pinnit	Micro:bit pinnit
A	Vihreä ledi	2	P0
B	Vastus 220Ω		

0. LEDIN VILKUTTELUA OHJEIDEN SOVELTAMINEN ERI LAITTEILLE

- Ohjeiden kuvat on piirretty vain Arduinolle mutta micro:bitille kytkentä on täysin vastaava, vain I/O pinnien numerointi on eri (ks. taulukko edellisellä sivulla).
- Lisäksi micro:bitiin liittyen ota huomioon, että
 - GND:n merkintä on Breakout boardissa 0V
 - Lähdejännite (V_s) on Arduinossa +5V mutta micro:bitissä +3V
 - Tarvitset F/M johtimia (Arduinon kanssa käytetään M/M johtimia)



Ledin vilkuttelua Arduinolla



Ledin vilkuttelua micro:bitillä

0. LEDIN VILKUTTELUA OHJELMOINTIYMPÄRISTÖN VALINTA

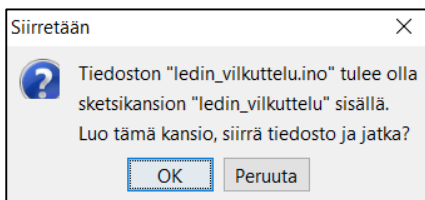
- Jos ohjelmoit Arduinoja asenna koneellesi [ArduinoCC](#)-ohjelmointiympäristö (tarjolla myös zip-tiedosto, joka ei vaadi asennusoikeuksia). ArduinoCC:llä ohjelmoidaan tekstipohjaisesti (C/C++). Jos tämä tuntuu liian haastavalta tai sinulla ei ole vielä käytössäsi Arduino-laitteita, voit ohjelmoida kuvakepohjaisesti selaimessa [TinkerCAD](#):illä. Ohjelman toiminnan voi testata selaimessa simuloinnin avulla. Jos koodi näyttää toimivan ok, sen voi ladata ArduinoCC:n avulla Arduinoon.
- Jos ohjelmoit micro:bitejä et tarvitse minkäänlaisia asennuksia. Valitse joko kuvakepohjainen selaimessa toimiva [Javascript Blocks](#) (jatkossa JSBlocks) tai tekstipohjainen (C++) [mbed micro:bit](#) (vaatii tilin luomisen).
- Lataa nyt valmis koodi laitteeseesi valitsemallasi ympäristöllä. Seuraa ohjeita:

Arduino	Micro:bit
ArduinoCC – ohje	JSBlocks – ohje
TinkerCAD – ohje	Mbed – ohje

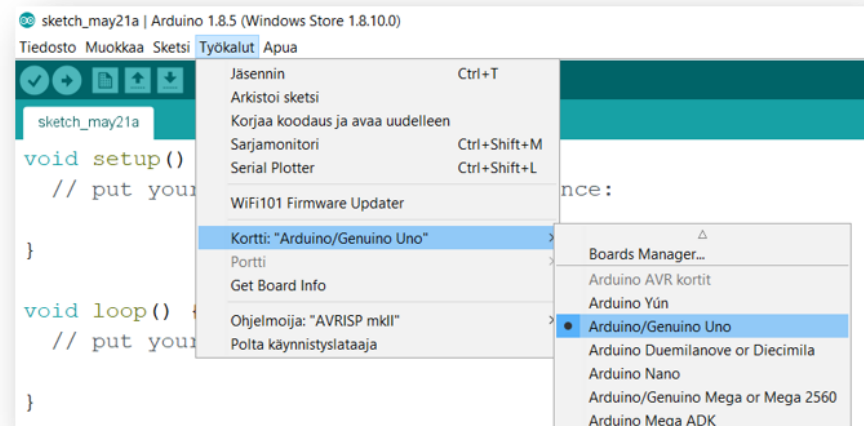
0. LEDINVILKUTTELUA ARDUINOCC



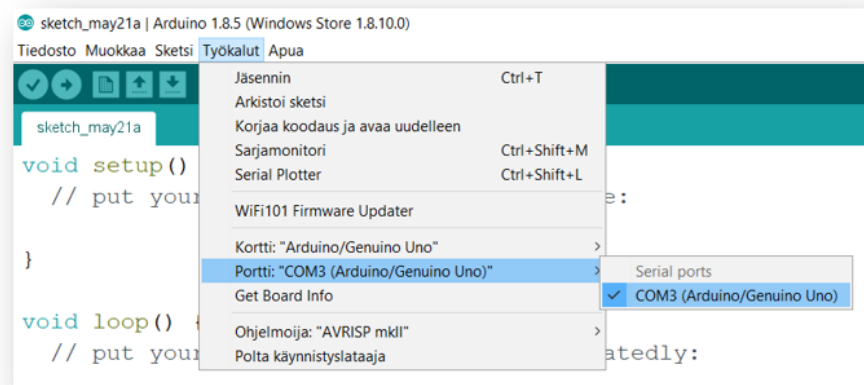
- Avaa valmis koodi [täältä](#) (tallenna .ino tiedosto ja avaa ArduinoCC-ohjelmassa). Vastaa OK kysyttäessä tehdäänkö sketsikansio (kuva 1).
- Kun olet ymmärtänyt mitä koodi tekee kiinnitä Arduino USB-kaapelilla tietokoneeseen.
- Tarkista, että ArduinoCC:ssä kortiksi on valittuna Arduino Uno (kuva 2)
- Valitse COM-portti, jossa Arduino on kiinni (kuva 3)
- Tarkista koodi painamalla 
- Lähetä laitteeseen painamalla 



Kuva 1



Kuva 2: Valitse kortiksi ArduinoUno

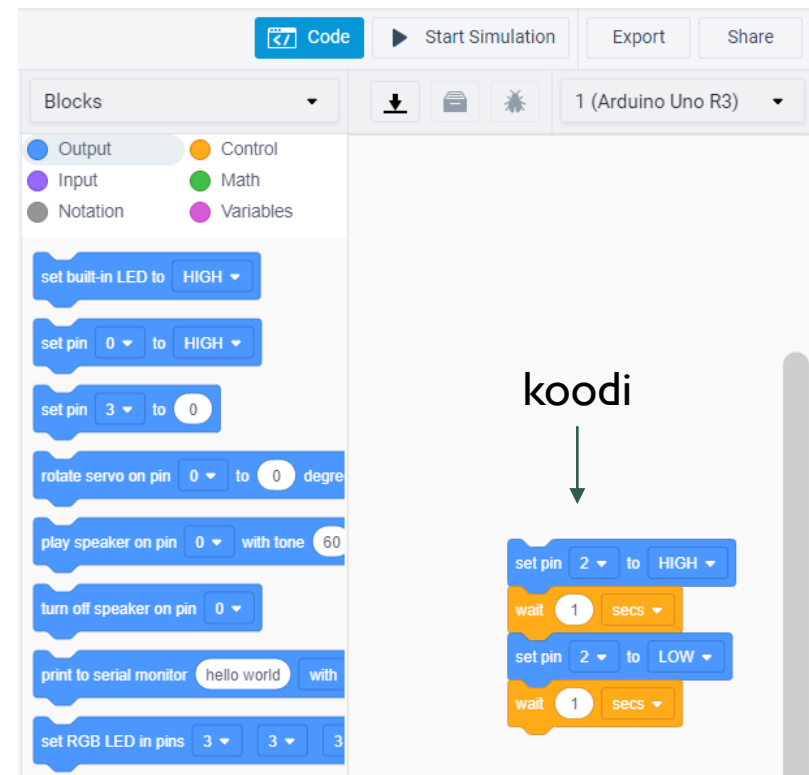


Kuva 3: Valitse COM-portti jossa laite on kiinni

0. LEDINVILKUTTELUA TINKERCAD



- Avaa valmis koodi [täältä](#) ja tutki sitä (koodi löytyy "Code" napin takaa).
- Tutki mitä koodi tekee painamalla "Start Simulation"
- Lopeta simulaatio ja lataa koodi tietokoneellesi painamalla 
- Kuvakkeilla tehdystä koodista generoituu ladattaessa C/C++ koodia, tallenna tiedosto (.ino) tietokoneellesi ja avaa ArduinoCC:llä
- Toimi kuten ArduinoCC:n kanssa (ks. edellinen sivu)
- Jos haluat tallentaa tai jakaa TinkerCAD:issä tekemäsi kytkennät/ koodin, tee sivustolle tunnukset

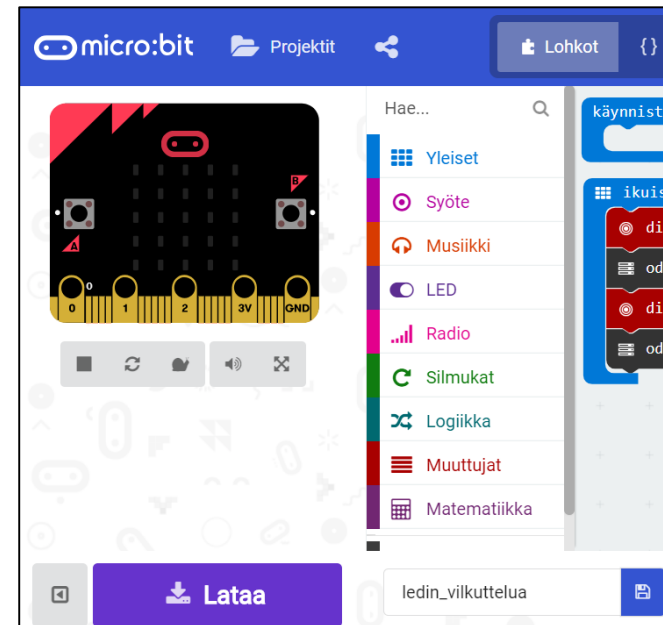


TinkerCAD:issä kaikki koodi on automaattisesti silmukan sisällä, yllä näkyvää koodia siis toistetaan ikuisesti

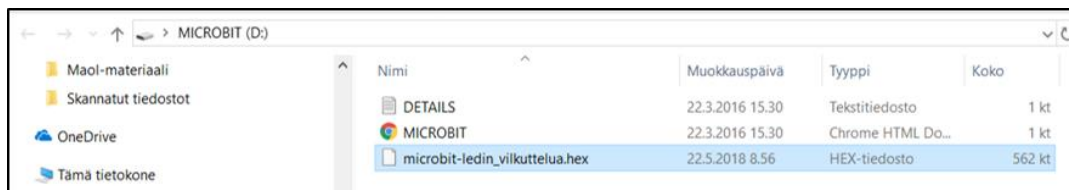
0. LEDIN VILKUTTELUA MICRO:BIT JS BLOCKS



- Avaa valmis koodi täältä.
- Kun olet ymmärtänyt mitä koodi tekee kiinnitä micro:bit USB-kaapelilla tietokoneeseen.
- Paina "Lataa" –nappia, jolloin kuvakkeista generoituu laitteen ymmärtämää koodia. Tallenna tiedosto (.hex) tietokoneelle. Avaa tiedoston sijaintikansio ja kopioi tiedosto micro:bitille, joka näkyy tiedoston hallinnassa kuin muistitikku esim. Microbit(D:).



Kuva 1: Lataa .hex tiedosto tietokoneellesi



Kuva 2: Siirrä .hex tiedosto micro:bitille

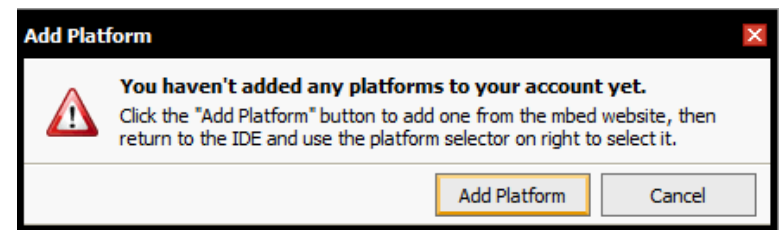
0. LEDIN VILKUTTELUA MICRO:BIT MBED



- Mbed-ympäristön käyttäminen vaatii käyttäjätilin luomisen. Mene [mbed:in pääsivulle](#) ja tee itsellesi tili (kuva 1), vahvista se sähköpostin avulla ja kirjaudu sisään.
- Ensimmäisellä kerralla mbed:iin pitää lisätä tieto ohjelmoitavasta laitteesta. Mene [micro:bitin sivulle](#) ja paina sivun oikeasta alalaidasta "Add to your mbed-compiler".
- Nyt voit avaa valmiin koodi [täältä](#). Valitse "Import into compiler". Etsi `main.cpp` tiedosto ja tutki koodia.
- Kun olet ymmärtänyt mitä koodi tekee kiinnitä micro:bit USB-kaapelilla tietokoneeseen ja paina "Compile". Nyt koodi kääntyy ajettavaa muotoon ja latautuu tietokoneellesi (tallenna tiedosto).
- Kopioi .hex-tiedosto micro:bittiin (ks. edellinen sivu).

A screenshot of the arm MBED login and signup form. It has a "Log in" tab and a "Sign up" tab. The "Log in" tab is active. There are input fields for "Email or username" and "Password", each with a "Forgotten your username?" or "Forgotten your password?" link below it. A yellow "Log In" button is at the bottom.

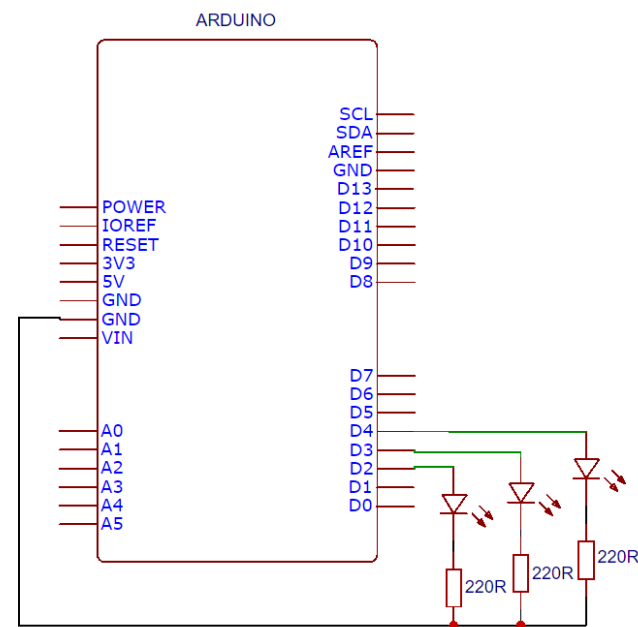
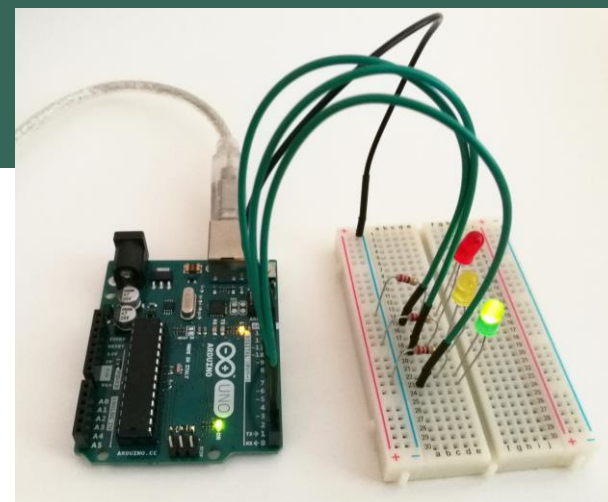
Kuva 1: Tee itsellesi käyttäjätili



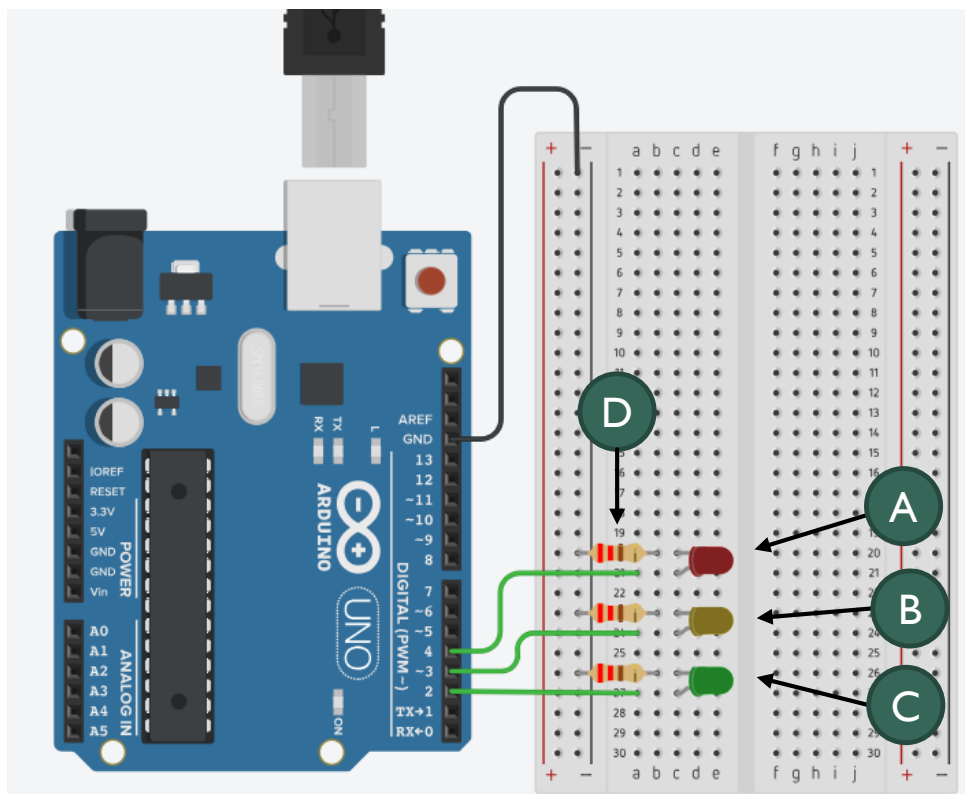
Kuva 2: Jos unohdat lisätä micro:bitin kääntäjään saat tämän virheilmoituksen

I. LEDIT LIIKENNEVALOINA

- Tässä työssä opitaan tarkemmin miten ledivaloja voidaan ohjata. Tutustu ledien ja vastusten käyttämiseen lukemalla niistä tietoisut (ks. linkit).
- Tehtäväsi on toteuttaa ledien avulla liikennevalot, jotka vaihtuvat seuraavasti: punainen – punainen ja keltainen – vihreä – keltainen – punainen jne.
- Tee kytkentä ohjeiden mukaisesti.
- Ohjelmointi:
 - Ledejä voidaan ohjata päälle (HIGH, 1) ja pois (LOW, 0) digitaalisen output-pinnin kautta (jokaisella ledillä on omansa)
 - Jotta saadaan ledi palamaan (tai pois päältä) tietyksi ajaksi, ohjelmoidaan odottamisaika (wait/delay/sleep)



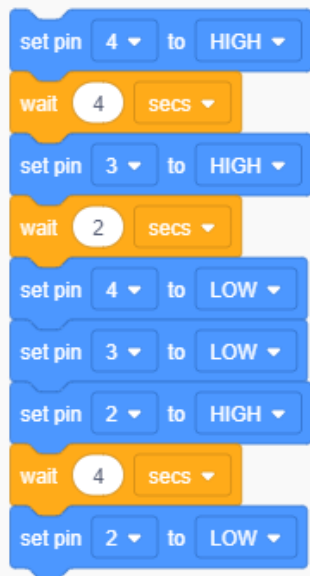
I. LEDIT LIIKENNEVALOINA



	Komponentit	Arduino pinnit	Micro:bit pinnit
A	Punainen ledi	4	P2
B	Keltainen ledi	3	P1
C	Vihreä ledi	2	P0
D	3x vastus 220Ω		

Arduino	Micro:bit
ArduinoCC	JSBlocks
TinkerCAD	mbed

I. ARDUINO-OHJELMOINTI



```
int LED_PUN = 4;  
int LED_KELT = 3;  
int LED_VIHR = 2;
```

```
void setup()  
{  
  pinMode(LED_PUN, OUTPUT);  
  pinMode(LED_KELT, OUTPUT);  
  pinMode(LED_VIHR, OUTPUT);  
}
```

```
void loop()  
{  
  digitalWrite(LED_PUN, HIGH);  
  delay(4000);  
  digitalWrite(LED_KELT, HIGH);  
  delay(2000);  
  digitalWrite(LED_PUN, LOW);  
  digitalWrite(LED_KELT, LOW);  
  digitalWrite(LED_VIHR, HIGH);  
  delay(4000);  
  digitalWrite(LED_VIHR, LOW);  
}
```

Muuttujien esittely ja alustaminen

Tässä esitellään ohjelmassa käytettävät muuttujat. Ensin ilmoitetaan muuttujan tyyppi `int` (integer, kokonaisluku). Muuttujaan sijoitetaan arvo = merkin avulla. Nämä muuttujat kuvaavat pinnejä, joihin ledit on kiinnitetty. Koska niitä ei muuteta ne ovat käytännössä vakioita ja kirjoitamme ne ISOLLA, jotta erotamme ne oikeasti muuttuvista muuttujista.

Alustus (setup)

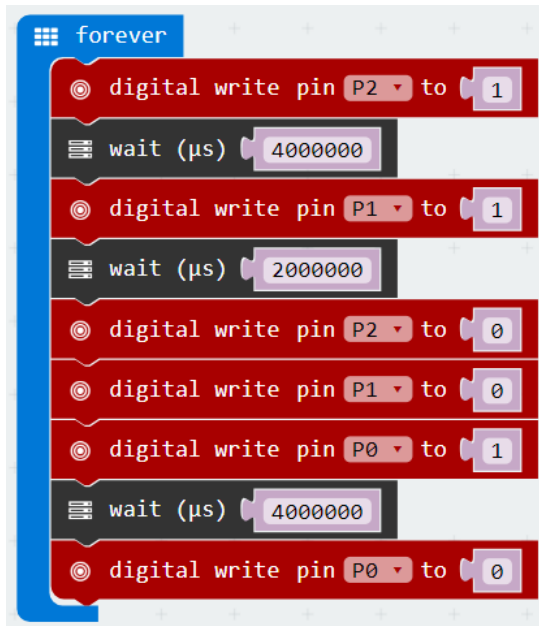
setup:in sisään kootaan kaikki koodi joka pitää ajaa ennen varsinaista ohjelmaa. Koska I/O pinnit voivat olla joko input tai output pinnejä, tässä asetetaan pinnien tyypit.

Silmukka (loop)

Tämän ikuisesti pyörivän silmukan sisälle kirjoitetaan Arduinossa suoritettava koodi. `digitalWrite` kirjoittaa pinniin ennetun arvon (HIGH/LOW). `delay` odottaa annetun ajan (ms) ja jatkaa seuraavalta riviltä.

Huom
TinkerCAD
käyttää
sekunteja (s) ja
ArduinoCC
millisekunteja
(ms).

I. MICRO:BIT-OHJELMOINTI



Pääohjelma (main)

C++ kielessä ohjelman suoritus alkaa pääohjelman sisältä eli `main`:istä.

```
#include "MicroBit.h"
MicroBit uBit;

int main()
{
    uBit.init();
    while (1){
        uBit.io.P2.setDigitalValue(1);
        uBit.sleep(4000);
        uBit.io.P1.setDigitalValue(1);
        uBit.sleep(2000);
        uBit.io.P2.setDigitalValue(0);
        uBit.io.P1.setDigitalValue(0);
        uBit.io.P0.setDigitalValue(1);
        uBit.sleep(4000);
        uBit.io.P0.setDigitalValue(0);
    }
}
```

Huom

JSBlocks käytetään mikrosekunteja (μ s) ja mbed:issä millisekunteja (ms)

Kirjasto ja microbit-olio

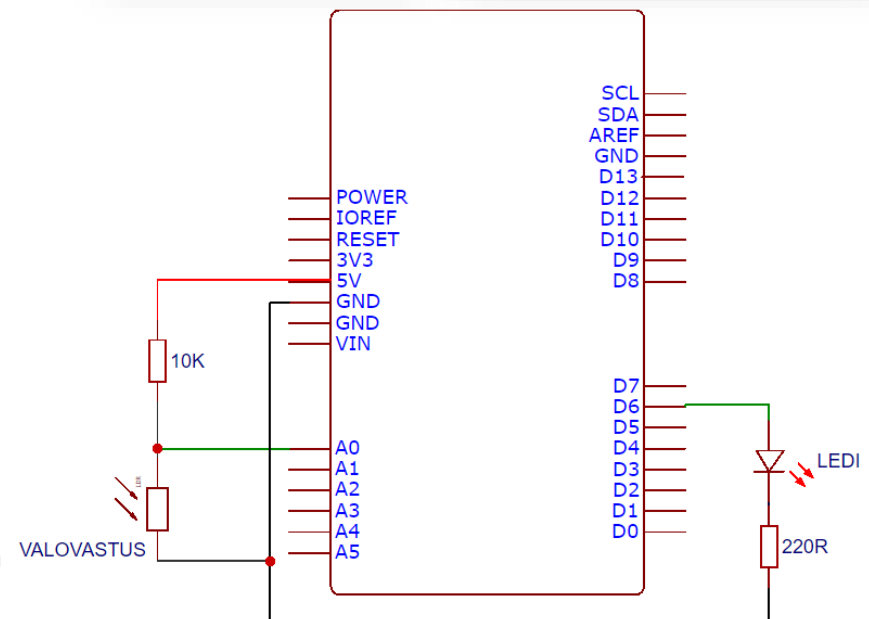
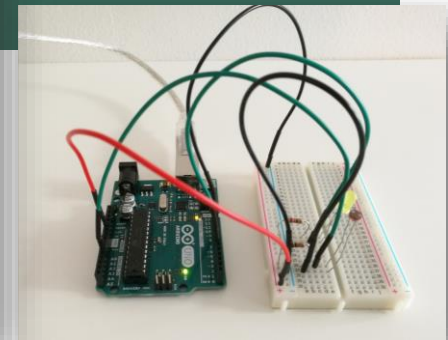
Mbed:issä ohjelmoidaan C++:lla joka on olio-ohjelmointikieli. Se tarkoittaa sitä, että käskyt annetaan ns. oliorajapinnan kautta (koodissa näkyvät pisteet liittyvät siihen). Jotta olioita voi käyttää tarvitaan kirjasto, jossa olion toiminta on ohjelmoituna (ensimmäinen koodirivi). Lisäksi pitää luoda yksi olio, tässä `MicroBit`-olio, johon viitataan jatkossa sille annetulla nimellä `uBit` (tämän saa itse valita). Ennen olion käyttöä se pitää alustaa kutsumalla `init()`-metodia.

Silmukka (while)

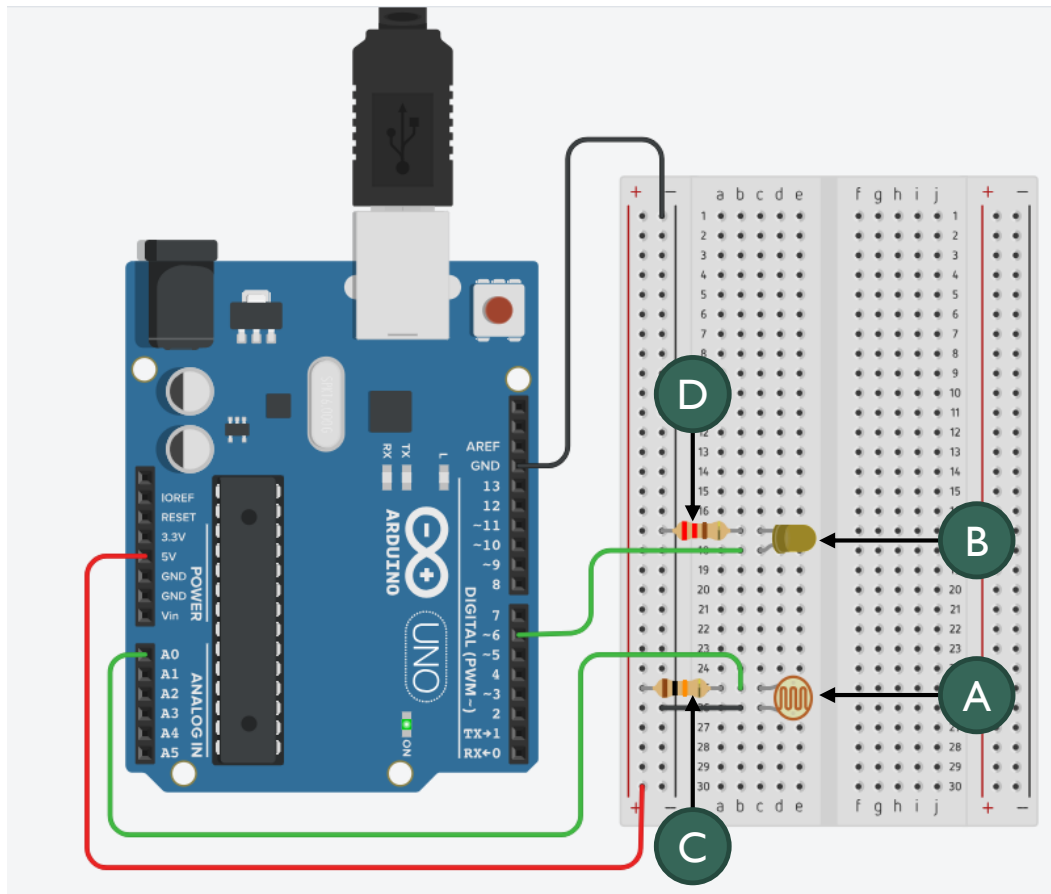
`while(1)` on ikuinen silmukka, joka toistaa sen sisälle kirjoitettua koodia. `uBit.io.P2.setDigitalValue(1)`-metodi kirjoittaa P2 pinniin annetun arvon (1). `sleep`-metodi odottaa annetun ajan (ms).

2.AUTOMAATTINEN LEDIMAJAKKA

- Jotta saamme automatisoitua toimintoja, laitteen täytyy osata aistia muutoksia ympäristössä. Majakan pitää alkaa vilkkumaan, kun tulee tarpeeksi pimeää ja sammua kun on jälleen riittävän valoisaa. Tätä varten tarvitsemme valosensorin tai valovastuksen. Tutustu näihin lukemalla niistä tietoiskut (ks. linkki).
- Tee kytkentä ohjeiden mukaisesti. Voit halutessasi myös rakentaa ”oikean” majakan talouspaperirullasta ja kirkkaasta muovisesta juomapullosta.
- Ohjelmointi:
 - Ensin luetaan valovastuksen/sensorin arvo analogisesta input pinnistä (välillä 0-1023)
 - Tutkitaan valovastuksen antamaa arvoa ehtolauseen (if-else) avulla ja valon määrästä riippuen aloitetaan ledin vilkuttaminen tai sammutetaan se kokonaan.
 - Vinkki: tulosta luettu valovastuksen arvo näkyville (Arduinossa sarjamonitorille, micro:bitissä ledinäytölle). Näin ehtolauseen vertailuoperaation säätäminen kohdilleen on helpompaa.



2.AUTOMAATTINEN LEDIMAJAKKA



	Komponentit	Arduino pinnit	Micro:bit pinnit
A	Valovastus/valosensori	A0	P1
B	Ledi	6	P0
C	Vastus 10kΩ		
D	Vastus 220Ω		

Arduino	Micro:bit
ArduinoCC	JSBlocks
TinkerCAD	mbed

2.ARDUIINO-OHJELMOINTI



```
int LED = 6;
int VALO = A0;
int arvo = 0;
```

```
void setup()
{
  pinMode(VALO, INPUT);
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}
```

```
void loop()
{
  arvo = analogRead(VALO);
  Serial.println(arvo);
  if (arvo <= 550) {
    digitalWrite(LED, LOW);
  } else {
    digitalWrite(LED, HIGH);
    delay(1000);
    digitalWrite(LED, LOW);
    delay(1000);
  }
}
```

Vertailuoperaatio

Vertailuoperaation palauttama arvo voi olla tosi tai epätosi. Jos se on tosi, ehtolauseen tosihaara suoritetaan, muuten suoritetaan epätosihaaran koodi.

Muuttujien esittely ja alustaminen

LED ja VALO ovat vakioita, jotka eivät muutu ohjelman aikana niihin tallennetaan I/O pinnien numerot. Muuttuja `arvo` on aito muuttuja, siihen tallennetaan ohjelman suorituksen aikana valovastukselta/-sensorilta luettuja arvoja (nolla tallennetaan alkuarvoksi).

Alustus (setup)

I/O pinnit asetetaan INPUT tai OUTPUT moodiin. Sarjamonitorin alustus (Serial.begin).

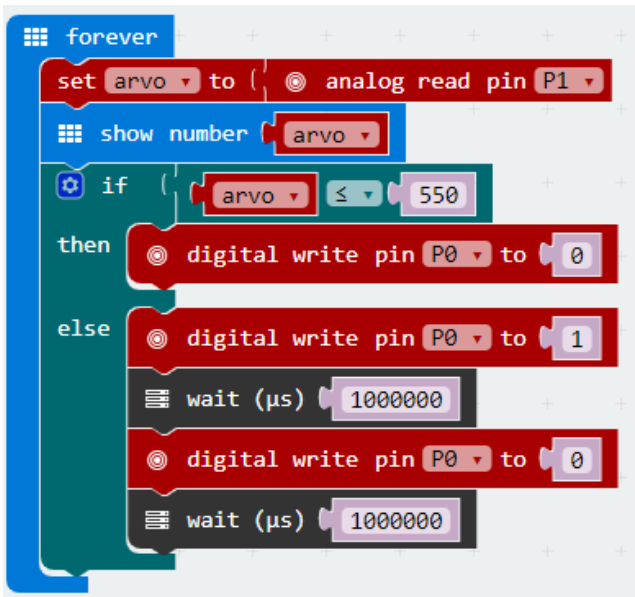
analogRead (funktio)

Tämän funktion avulla luetaan INPUT pinnin (VALO) jännitearvo. Saatu arvo tallennetaan muuttujaan `arvo`. Jotta ohjelmointi helpottuu tulostetaan `arvo` myös sarjamonitorille. Sarjamonitorin saa auki valikosta *Työkalut* → *Sarjamonitori*.

Ehtolause (if-else)

Ehtolauseen avulla saadaan tehtyä ehdollista koodia: jos (`if`) valovastuksen antama lukema on pienempi tai yhtä suuri kuin 550, sammutetaan ledi, muuten (`else`) vilkutetaan lediä sekunnin välein.

2. MICRO:BIT-OHJELMOINTI



Ehtolause (if-else)

Ehtolauseen avulla saadaan tehtyä ehdollista koodia: jos (if) valovastuksen jännite on pienempi tai yhtä suuri kuin 550, sammutetaan ledi, muuten (else) vilkutetaan lediä sekunnin välein.

```
#include "MicroBit.h"

MicroBit uBit;
int arvo = 0;

int main()
{
    uBit.init();
    while (1)
    {
        arvo = uBit.io.P1.getAnalogValue();
        ManagedString arvo_str(arvo);
        uBit.display.scrollAsync(arvo_str);
        if(arvo <= 550)
        {
            uBit.io.P0.setDigitalValue(0);
        }
        else
        {
            uBit.io.P0.setDigitalValue(1);
            uBit.sleep(1000);
            uBit.io.P0.setDigitalValue(0);
            uBit.sleep(1000);
        }
    }
}
```

Muuttujat

Otetaan käyttöön kokonaislukutyypinen (int) muuttuja `arvo` ja annetaan sille alkuarvoksi nolla. Tähän muuttujaan tallennetaan ohjelman aikana luetut valovastuksen jännitearvot.

getAnalogValue

Tämän metodin avulla luetaan P1-pinnin arvo. Se tallennetaan muuttujaan `arvo`.

scrollAsync

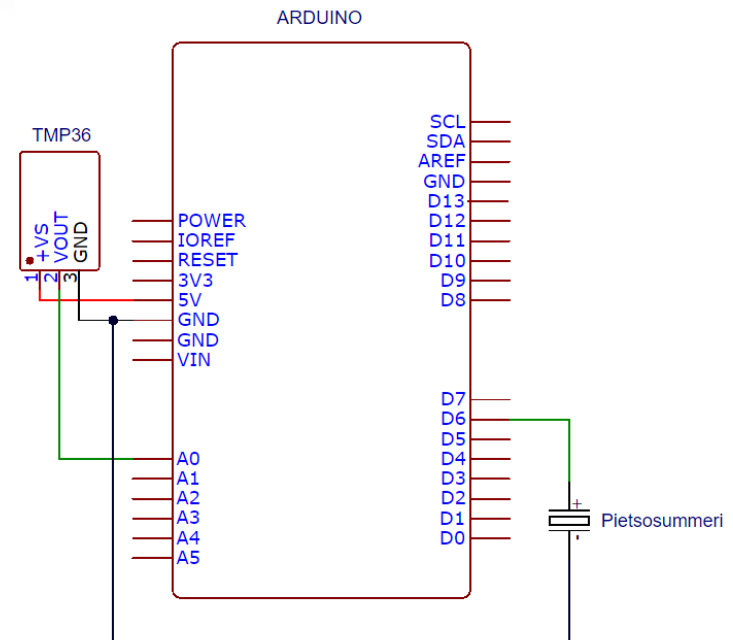
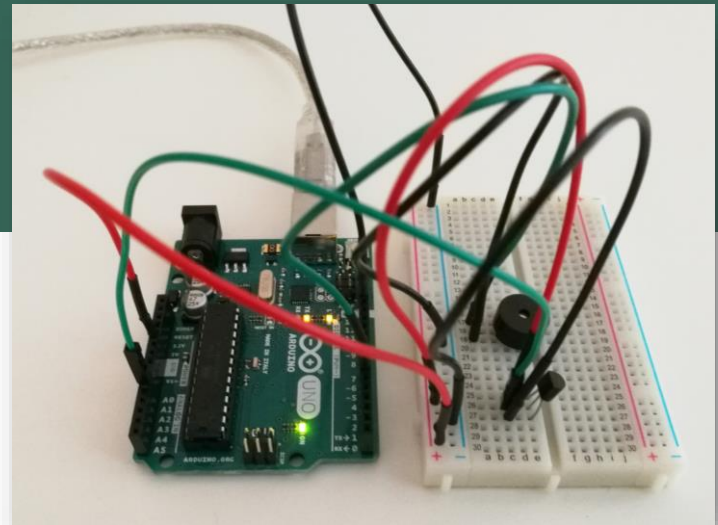
Tämän metodin avulla tulostetaan arvo näytölle. Koska se on tyyppi `int`, se pitää ensin muuttaa merkkijonoksi (`ManagedString`).

Vertailuoperaatio

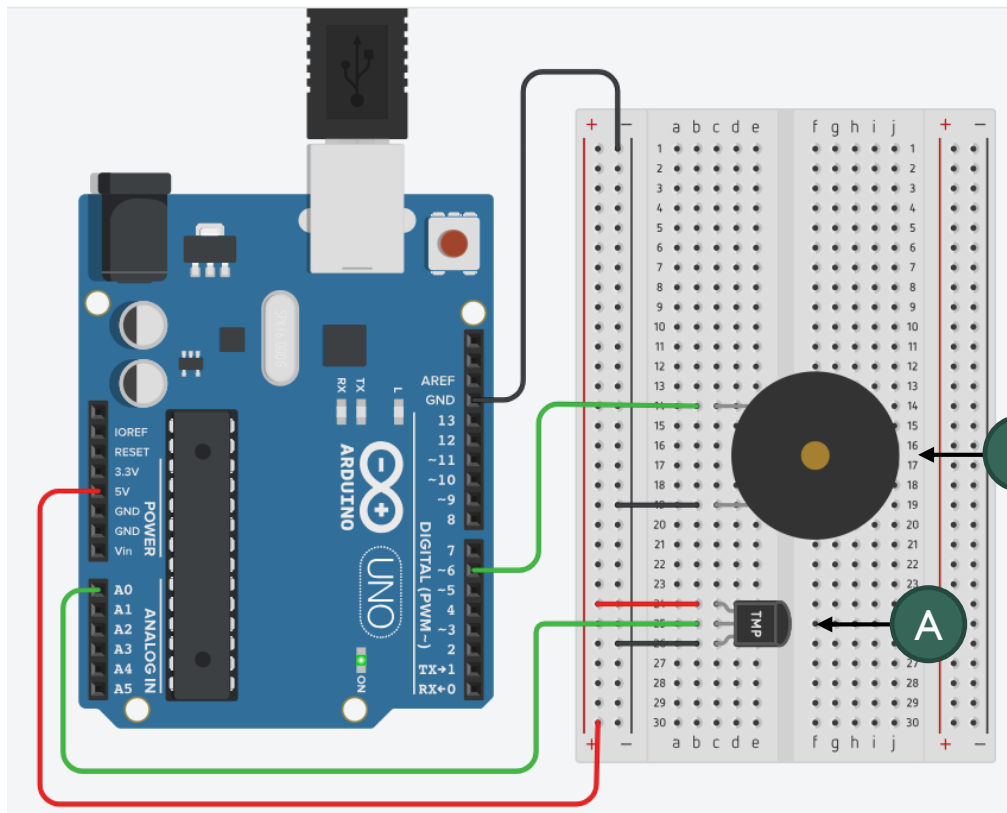
Vertailuoperaation palauttama arvo voi olla tosi tai epätosi. Jos se on tosi, ehtolauseen tosihaara suoritetaan, muuten suoritetaan epätosihaaran koodi.

3. LÄMPÖTILAVAHTI

- Tässä työssä opetellaan käyttämään lämpötilasensoria sekä pietsosummeria. Näiden komponenttien avulla saadaan toteutettua liian korkeasta (tai matalasta) lämpötilasta hälyttävä lämpötilavahti. Tutustu komponentteihin lukemalla niistä tietoiskut (ks. linkit).
- Tee kytkentä ohjeiden mukaisesti. Huomaa, että micro:bitin kanssa summeri hälyttää hiljempaa, koska sen antama jännite on vain 3V. **Tarkista, että kytket lämpötilasensorin oikein!**
- Ohjelmointi:
 - Ensin luetaan lämpötilasensorin arvo analogisesta input pinnistä (välillä 0-1023)
 - Luettu jännitearvo muunnetaan lämpötilaksi (°C) laskukaavan mukaisesti (ks. lämpötilasensorin tietoisku)
 - Ehtolauseen avulla lämpötilasta päätellään pitääkö olla hiljaa vai soittaa hälytys-summeria (C2-säveltä eli 523Hz)



3. LÄMPÖTILAVAHTI

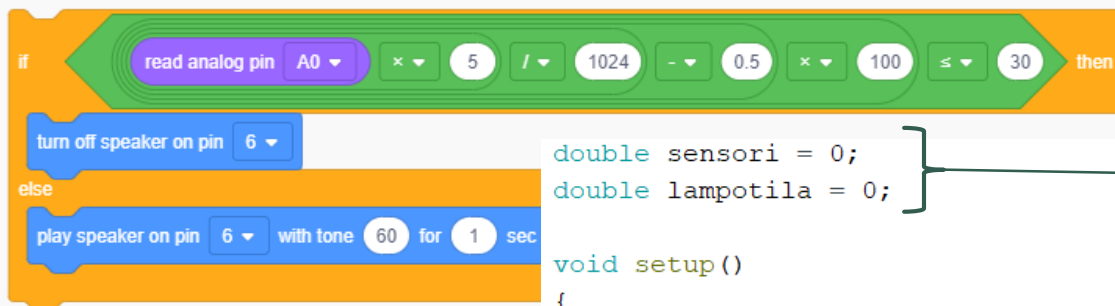


	Komponentit	Arduino pinnit	Micro:bit pinnit
A	Lämpötila-sensori TMP36 *)	A0	P1
B	Pietso-summeri	~6	P0

*) voit käyttää muitakin sensoreita (esim. LM35) mutta muista silloin vaihtaa sensorin lämpötilan muunnoskaava koodissa

Arduino	Micro:bit
ArduinoCC	JSBlocks
TinkerCAD	mbed

3. ARDUINO-OHJELMOINTI



Alustus (setup)

Otamme käyttöön sarjamonitorin, jonka avulla voi debugata.

Sarjamonitori

Nämä rivit kirjoittavat sarjamonitorille muuttujien arvoja. Sarjamonitorin saa auki valikosta *Työkalut* → *Sarjamonitori*

Tone

tone-funktio soittaa pietsosummerilla 523Hz ääntä sekunnin ajan.

```
double sensori = 0;
double lampotila = 0;

void setup()
{
  pinMode(A0, INPUT);
  pinMode(6, OUTPUT);
  Serial.begin(9600);
}

double muunna_lampotilaksi(double arvo)
{
  return ((arvo * 5) / 1024 - 0.5) * 100;
}

void loop()
{
  sensori = analogRead(A0);
  lampotila = muunna_lampotilaksi(sensori);
  Serial.println(sensori);
  Serial.println(lampotila);

  if (lampotila <= 30) {
    noTone(6);
  } else {
    tone(6, 523, 1000);
  }
  delay(10);
}
```

Muuttujien esittely ja alustaminen

Käytettävät apumuuttujat pitää määritellä liukuluvuiksi (double), koska käytämme niitä jakolaskuissa. int-tyyppisten muuttujien käyttö aiheuttaisi lukujen pyöristymisen kokonaisluvuiksi.

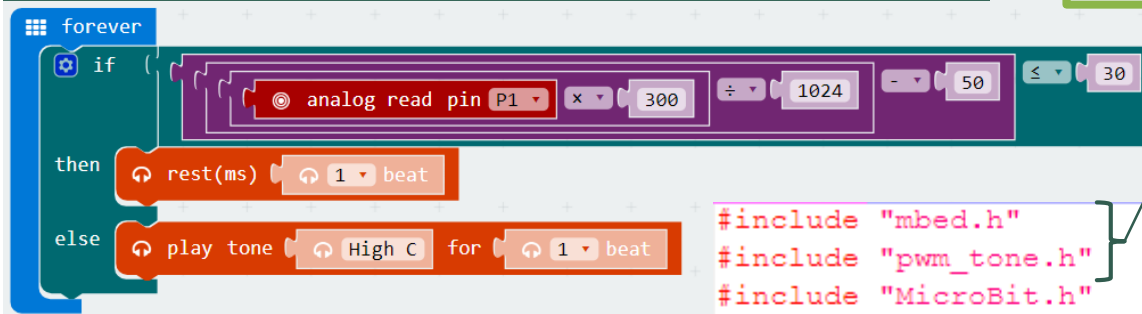
Funktion määrittely

Teemme apufunktion muunna_lampotilaksi joka saa parametrinä sensorin arvon, jonka se muuntaa laskukaavan avulla lämpötilaksi. Saatu lämpötila palautetaan (return) kutsujalle. Rivin ensimmäinen double viittaa palautettavan tiedon tyyppiin, toinen double parametrinä saatavan tiedon tyyppiin.

Funktiokutsu

Vasta tässä käytetään määriteltyä apufunktiota. Funktion laskema lämpötila tallennetaan muuttujaan lampotila.

3. MICRO:BIT-OHJELMOINTI



Funktion määrittely

Teemme apufunktion `muunna_lampotilaksi` joka saa parametrinä sensorin arvon jonka se muuntaa laskukaavan avulla lämpötilaksi. Saatu lämpötila palautetaan (`return`) kutsujalle. Rivin ensimmäinen `double` viittaa palautettavaan tiedon tyyppiin, toinen `double` parametrinä saatavan tiedon tyyppiin.

Funktiokutsu

Vasta tässä käytetään määriteltyä apufunktiota. Funktion laskema lämpötila tallennetaan muuttujaan `lampotila`.

Kirjastot

Pietsosummerin käyttö vaatii `mbed.h` ja `pwm_tone.h` kirjastot.

Muuttujien esittely ja alustaminen

Otetaan käyttöön kaksi apumuuttujaa `sensori` ja `lampotila`. Ne pitää määrittellä liukuluvuiksi (`double`), koska käytämme niitä jakolas-kuissa. `int`-tyyppisten muuttujien käyttö aiheuttaisi lukujen pyöristymisen kokonaisluvuiksi.

```
#include "mbed.h"
#include "pwm_tone.h"
#include "MicroBit.h"

MicroBit uBit;
* PwmOut Buzzer(MICROBIT_PIN_P0);
float c_2 = 1000000/523;
double sensori = 0;
double lampotila = 0;

double muunna_lampotilaksi(double arvo){
    return ((arvo * 3) / 1024 - 0.5) * 100;
}

int main(){
    uBit.init();
    while (1){
        sensori = uBit.io.P1.getAnalogValue();
        lampotila = muunna_lampotilaksi(sensori);
        if(lampotila <= 30){
            Stop_tunes(Buzzer);
        } else{
            Tune(Buzzer, c_2, 8);
            wait_ms(250);
        }
    }
}
```

* PwmOut

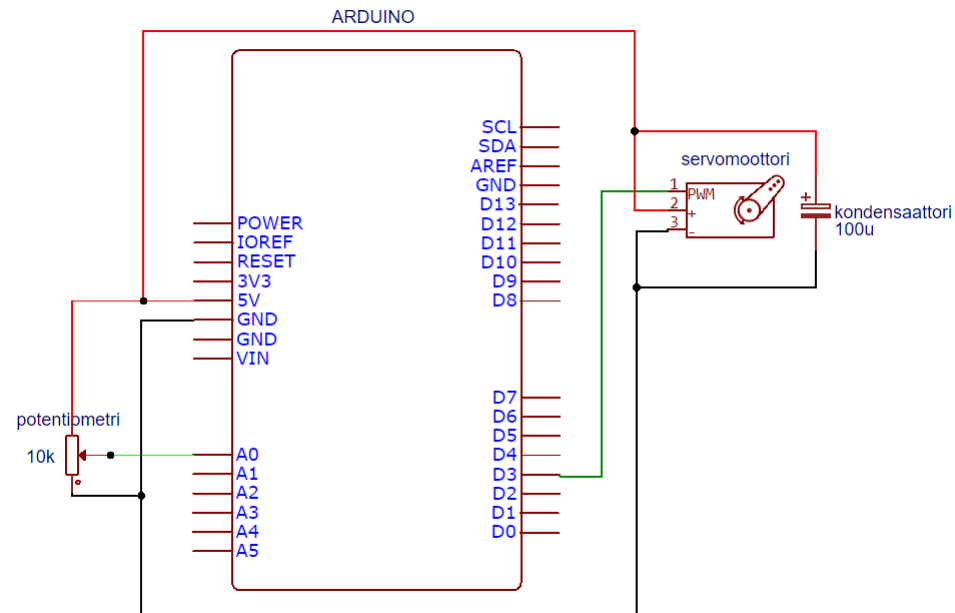
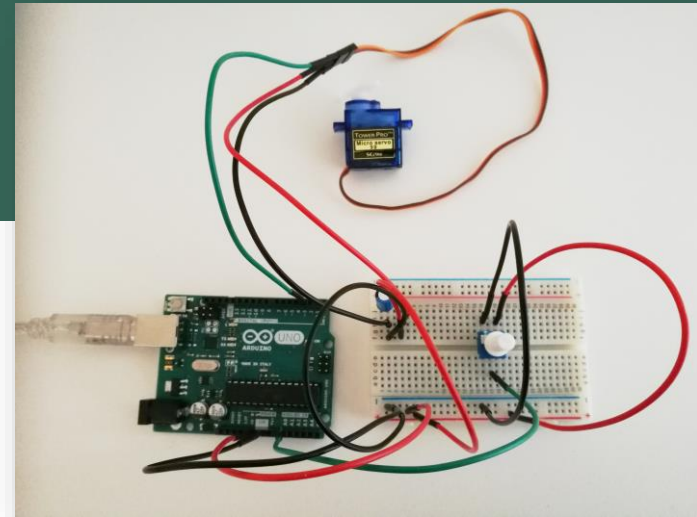
Pietsosummerin ohjaaminen tapahtuu `PwmOut`-objektin (`Buzzer`) avulla.

Tune

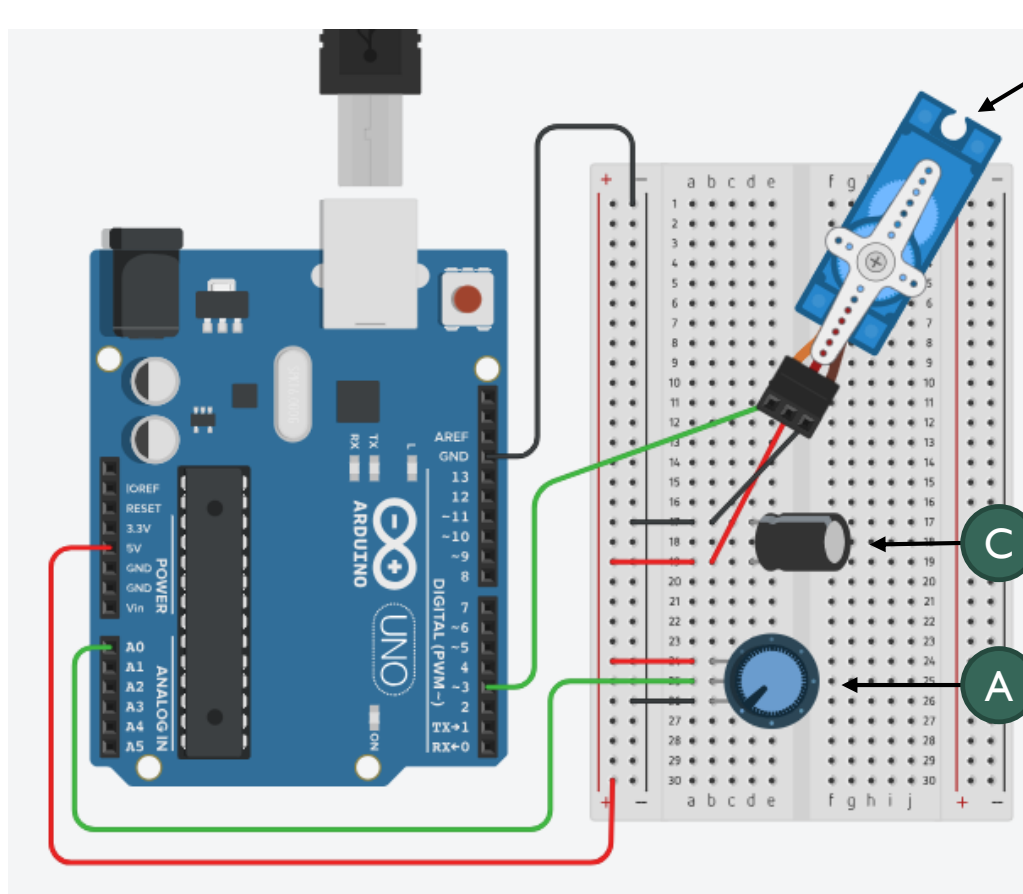
`tune`-funktio soittaa annettua ääntä (`c_2`) puolinuotin (8/16) ajan.

4. POTENTIOMETRILLÄ OHJATTU SERVO

- Tässä työssä opitaan miten servomoottori saadaan reagoimaan potentiometriä kääntämällä. Tutustu komponentteihin lukemalla niistä tietoiskut (ks. linkit).
- Tee kytkentä ohjeiden mukaisesti. Kondensaattorin voi jättää pois mutta silloin servomoottori saattaa nykiä.
- Ohjelmointi:
 - Ensin luetaan potentiometrin arvo analogisesta input pinnistä (välillä 0-1023)
 - Luettu arvo muunnetaan servomoottorin asteiksi (välille 0-180) ja kirjoitetaan analogiseen output pinniin



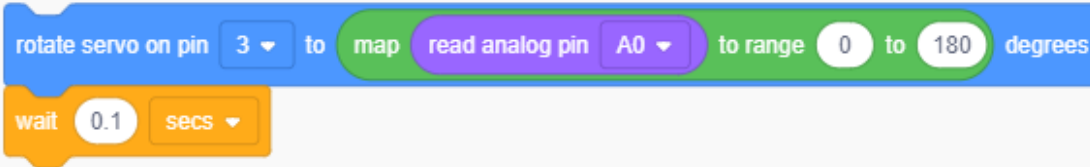
4. POTENTIOMETRILLÄ OHJATTU SERVOMOOTTORI



	Komponentit	Arduino pinnit	Micro:bit pinnit
A	Potentio- metri	A0	P1
B	Servo- moottori	~3	P0
C	Konden- saattori		

Arduino	Micro:bit
ArduinoCC	JSBlocks
TinkerCAD	mbed

4. ARDUINO-OHJELMOINTI



Kirjastot

Servomoottorin ohjaaminen vaatii `<Servo.h>` kirjaston käyttöön ottamisen. Tämä kirjasto löytyy valmiina ArduinoCC:stä.

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

```
int SERVO = 3;
int POT = A0;
int potikka = 0;
int servo = 0;
Servo servo_m;
```

```
void setup()
{
  servo_m.attach(SERVO);
}
```

```
void loop()
{
  potikka = analogRead(POT);
  servo = map(potikka, 0, 1023, 0, 180);
  servo_m.write(servo);
  delay(100);
}
```

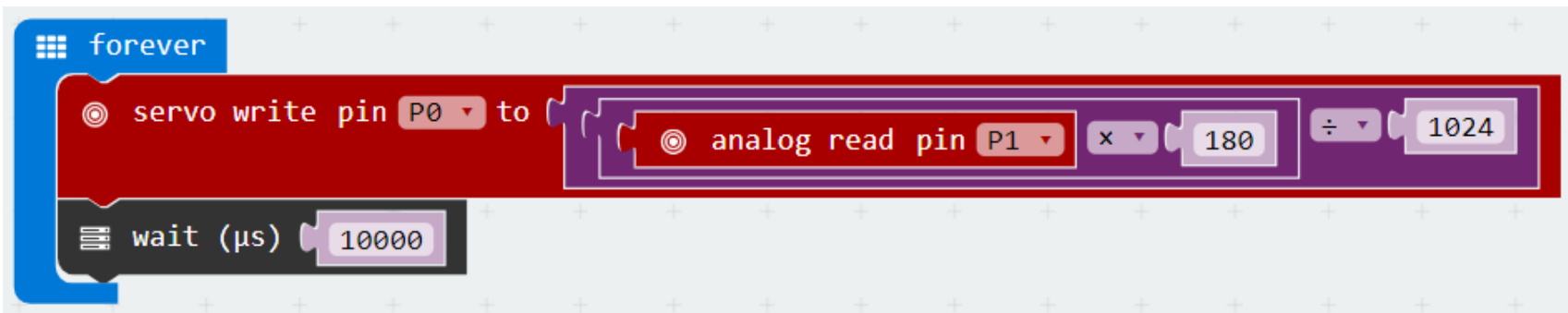
Muuttujien esittely ja alustaminen

Annetaan I/O-pinneille nimet SERVO ja POT. Luodaan apumuuttujat potikka ja servo ja annetaan niille alkuarvoksi nolla. Luodaan Servo-olio servo_m, jonka avulla moottoria ohjailaan. Alustetaan Servo-olio antamalla sen pinnin numero, missä servomoottori on kiinni

AnalogRead, map, delay

analogRead palauttaa arvon välillä 0-1023, ja se tallennetaan muuttujaan potikka. Servon ohjaaminen tapahtuu kuitenkin asteina välillä 0-180, siksi potikan arvo pitää skaalata käyttäen map-funktiota. map-funktiolle annetaan lähtöalue (0-1023) ja kohdealue (0-180). servo_m.write ohjaa moottorin kääntymään. Pieni delay on tarpeen, koska servo reagoi pienellä viiveellä.

4. MICRO:BIT-OHJELMOINTI



```
#include "MicroBit.h"

MicroBit uBit;
int potikka = 0;
int servo = 0;

int main()
{
    uBit.init();

    while(1)
    {
        potikka = uBit.io.P1.getAnalogValue();
        servo = (int)(potikka/1024.0*180);
        uBit.io.P0.setServoValue(servo);
        uBit.sleep(100);
    }
}
```

Kirjasto ja microbit-olio

Otetaan käyttöön `MicroBit.h` -kirjasto ja luodaan `MicroBit`-olio.

Muuttujien esittely ja alustaminen

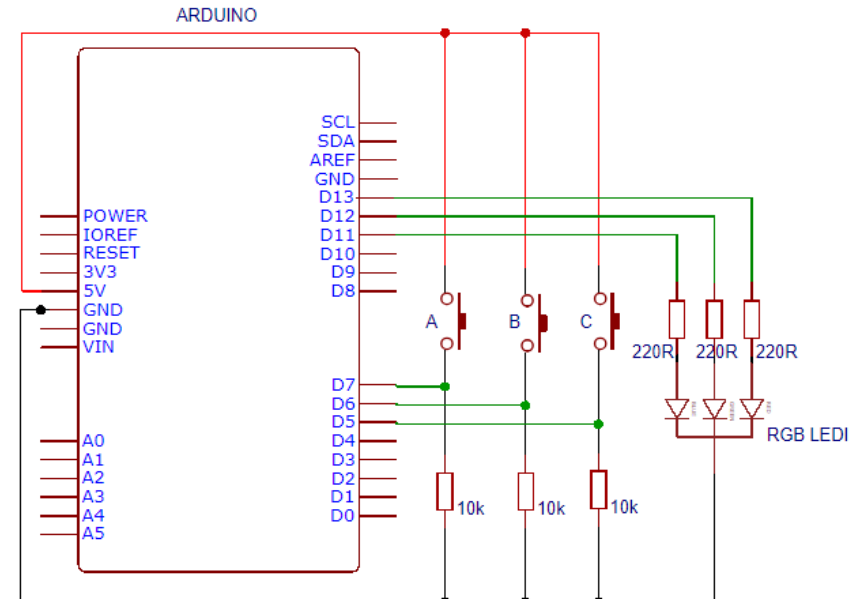
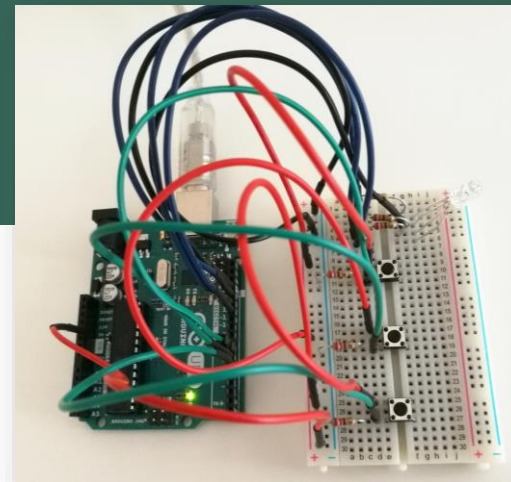
Luodaan apumuuttujat `potikka` ja `servo` ja annetaan niille alkuarvoksi nolla. Alustetaan `MicroBit`-olio.

getAnalogValue, setServoValue, sleep

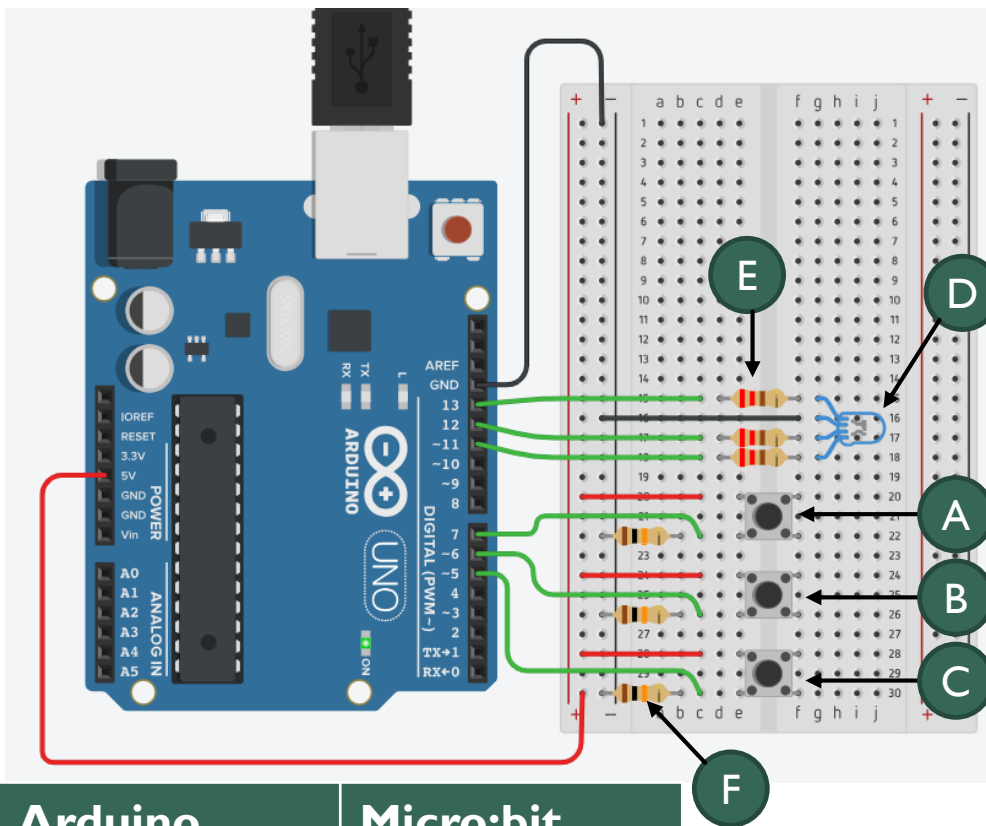
Luetaan potentiometrin arvo `getAnalogValue`-metodin avulla ja tallennetaan se muuttujaan `potikka`. Servon ohjaaminen tapahtuu kuitenkin asteina välillä 0-180, siksi potikan arvo pitää skaalata välille 0-180. Skaalattu arvo tallennetaan muuttujaan `servo`. Ohjataan moottori kääntymään halutun verran `setServoValue`-metodin avulla. Odotetaan 100ms, että moottori ehtii reagoida (`sleep`).

5. VÄRIVALOJEN OHJAUSPANELI

- Tässä työssä tutustutaan painokytkimiin sekä RGB-lediin. Kytkimien avulla ohjataan ledi palamaan eri värisenä. Tutustu komponentteihin lukemalla niistä tietoiskut (ks. linkit).
- Tee kytkentä ohjeiden mukaisesti. Jos sinulla ei ole RGB-lediä (koostuu kolmesta erivärisestä ledistä), voit korvata sen punaisella, vihreällä ja sinisellä ledillä.
- Ohjelmointi:
 - Kun painetaan kytkintä A, sytytetään RGB-ledin punainen ledi. Kun painetaan kytkintä B, sytytetään RGB-ledin vihreä ledi ja kytkintä C RGB-ledin sininen ledi. Kun kaikkia kytkimiä painetaan yhtä aikaa valot alkavat vaihtua automaattisesti sekunnin välein syklissä: punainen-vihreä-sininen.



5.VÄRIVALOJEN OHJAUSPANELI



Arduino	Micro:bit
<u>ArduinoCC</u>	<u>JSBlocks</u>
<u>TinkerCAD</u>	<u>mbed</u>

	Komponentit	Arduino pinnit	Micro:bit pinnit
A	Painonappi	7	P1
B	Painonappi	6	P5
C	Painonappi	5	P0
D	RGB-LED (punainen)	13	P2
	RGB-LED (vihreä)	12	P11
	RGB-LED (sininen)	11	P8
E	3x vastus 220Ω		
F	3x vastus 10kΩ		

5. ARDUINO-OHJELMOINTI

set A ▼ to read digital pin 7 ▼
set B ▼ to read digital pin 6 ▼
set C ▼ to read digital pin 5 ▼

if (A == 1 and B == 1 and C == 1) then

set pin 13 ▼ to HIGH ▼

wait 1 secs ▼

set pin 13 ▼ to LOW ▼

set pin 12 ▼ to HIGH ▼

wait 1 secs ▼

set pin 12 ▼ to LOW ▼

set pin 11 ▼ to HIGH ▼

wait 1 secs ▼

set pin 11 ▼ to LOW ▼

else

if (A == 1) then

set pin 13 ▼ to HIGH ▼

else

set pin 13 ▼ to LOW ▼

if (B == 1) then

set pin 12 ▼ to HIGH ▼

else

set pin 12 ▼ to LOW ▼

if (C == 1) then

set pin 11 ▼ to HIGH ▼

else

set pin 11 ▼ to LOW ▼

```
int A = 0;
int B = 0;
int C = 0;

void setup() {
  pinMode(7, INPUT);
  pinMode(6, INPUT);
  pinMode(5, INPUT);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}
```

Sisäkkäiset ehtolauseet

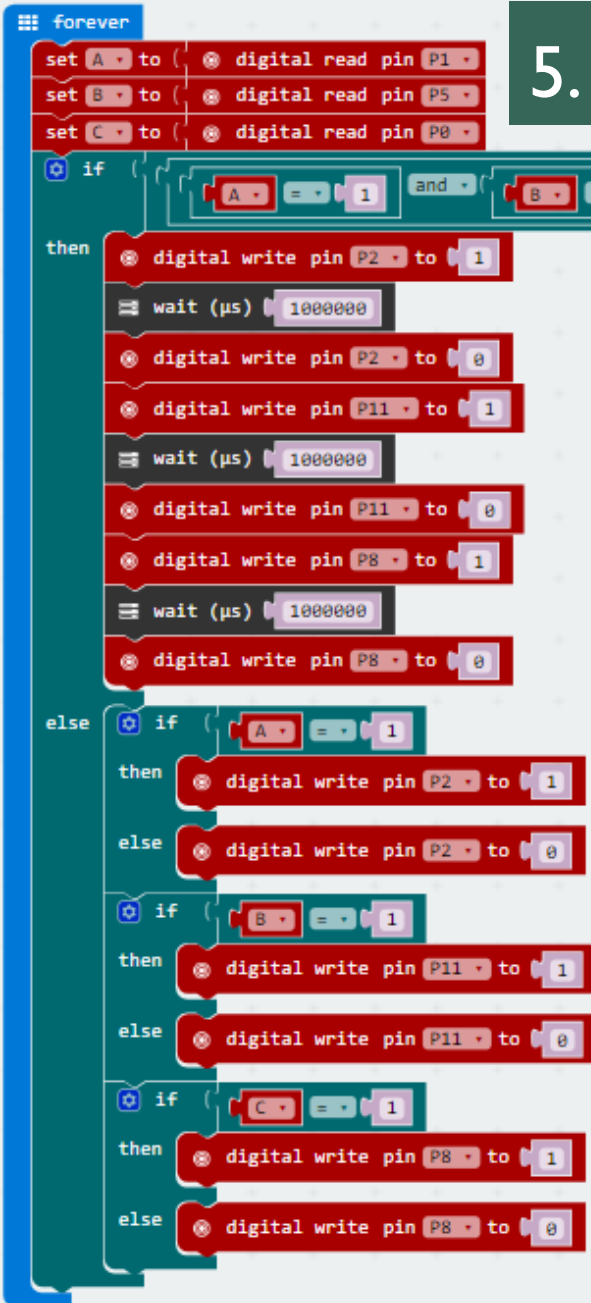
Tässä uloimman ehtolauseen epätosihaaran sisällä on kolme erillistä ehtolauseetta. Jos uloimman ehtolauseen tosihaara suoritetaan, tämän epätosihaaran yli hypätään kokonaan.

```
void loop() {
  A = digitalRead(7);
  B = digitalRead(6);
  C = digitalRead(5);
  if ((A == 1 && B == 1) && C == 1) {
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    digitalWrite(12, HIGH);
    delay(1000);
    digitalWrite(12, LOW);
    digitalWrite(11, HIGH);
    delay(1000);
    digitalWrite(11, LOW);
  } else {
    if (A == 1) {
      digitalWrite(13, HIGH);
    } else {
      digitalWrite(13, LOW);
    }
    if (B == 1) {
      digitalWrite(12, HIGH);
    } else {
      digitalWrite(12, LOW);
    }
    if (C == 1) {
      digitalWrite(11, HIGH);
    } else {
      digitalWrite(11, LOW);
    }
  }
}
```

Boolean operaattorit and ja or

Tässä on rakennettu ehtolause yhdistelemällä kolme erillistä ehtoa *and* (&&) operaattorin avulla: "jos A on 1 ja jos B on 1 ja jos C on 1 niin...". Merkintä == tarkoittaa "on yhtä kuin". Jos riittää että yksikin ehdoista on totta yhdistä, ehdot käyttämällä *or* (||) operaattoria.

5. MICROBIT-OHJELMOINTI



```
#include "MicroBit.h"

MicroBit uBit;
int A = 0;
int B = 0;
int C = 0;
```

Boolean operaattorit and ja or

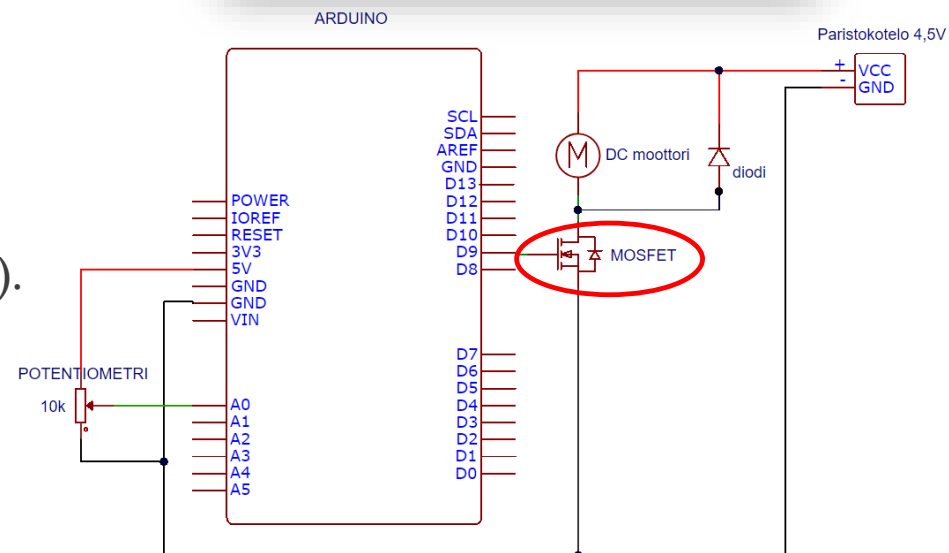
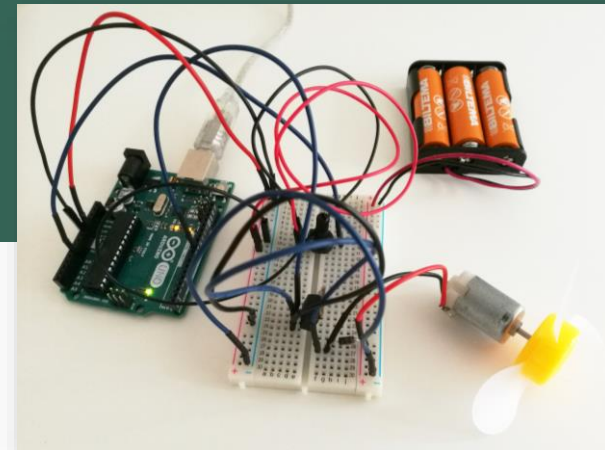
Tässä on rakennettu ehtolause yhdistelemällä kolme erillistä ehtoa *and* (&&) operaattorin avulla: "jos A on 1 ja jos B on 1 ja jos C on 1 niin...". Merkintä == tarkoittaa "on yhtä kuin". Jos riittää että yksikin ehdoista on totta yhdistä, ehdot käyttämällä *or* (||) operaattoria.

```
int main() {
    uBit.init();
    while (1) {
        A = uBit.io.P1.getDigitalValue();
        B = uBit.io.P5.getDigitalValue();
        C = uBit.io.P0.getDigitalValue();

        if ((A == 1 && B == 1) && C == 1) {
            uBit.io.P2.setDigitalValue(1);
            uBit.sleep(1000);
            uBit.io.P2.setDigitalValue(0);
            uBit.io.P11.setDigitalValue(1);
            uBit.sleep(1000);
            uBit.io.P11.setDigitalValue(0);
            uBit.io.P8.setDigitalValue(1);
            uBit.sleep(1000);
            uBit.io.P8.setDigitalValue(0);
        } else {
            if(A == 1){
                uBit.io.P2.setDigitalValue(1);
            } else {
                uBit.io.P2.setDigitalValue(0);
            }
            if(B == 1){
                uBit.io.P11.setDigitalValue(1);
            } else {
                uBit.io.P11.setDigitalValue(0);
            }
            if(C == 1){
                uBit.io.P8.setDigitalValue(1);
            } else {
                uBit.io.P8.setDigitalValue(0);
            }
        }
    }
}
```

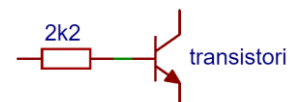
6. POTENTIOMETRILLÄ OHJATTU DC-MOOTTORI

- Tässä työssä tutustutaan DC-moottorin ohjaamiseen transistorin/MOSFET:in, diodin ja potentiometrin avulla. Tutustu komponentteihin lukemalla niistä tietoiskut (ks. linkit).
- Valitse kumman kytkennän teet: MOSFET (ei toimi micro:bitillä) vai transistori. Huomaa, että moottorilla on oma jännitelähteensä (paristot), jonka miinusnapa kytketään Arduinon/micro:bitin kanssa samaan maahan (GND). Huom! Pariston plusnapaa EI kytketä Arduinon/micro:bitiin!
- Ohjelmointi:
 - Luetaan potentiometrin antama arvo (0-1023). Tämän arvon avulla ohjataan moottorin pyörimisnopeutta, mitä suurempi jännitteen arvo sitä nopeammin moottori pyörii. Huom. Arduinon kanssa potentiometrin antama arvo pitää skaalata välille 0-255 (map).

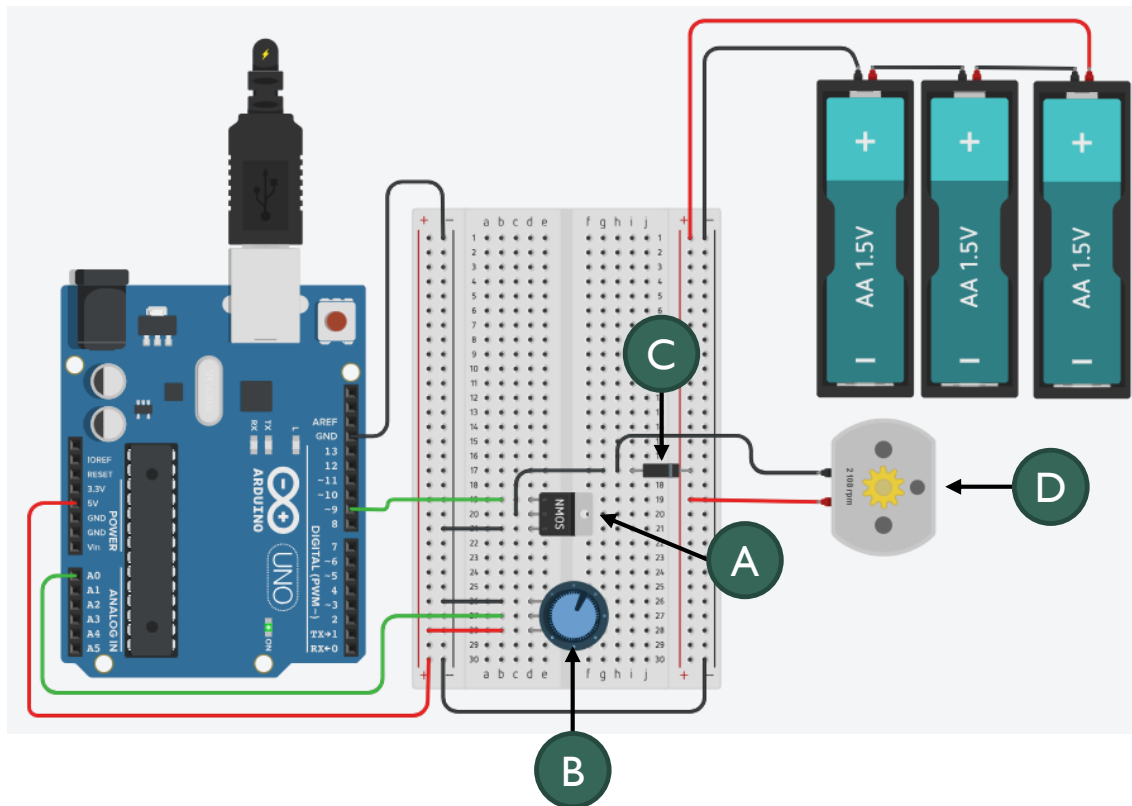


Mosfet-kytkentä

Transistorikytkennässä MOSFET:in tilalle laitetaan vastus ja transistori



6. POTENTIOMETRILLÄ OHJATTU DC-MOOTTORI (MOSFET)



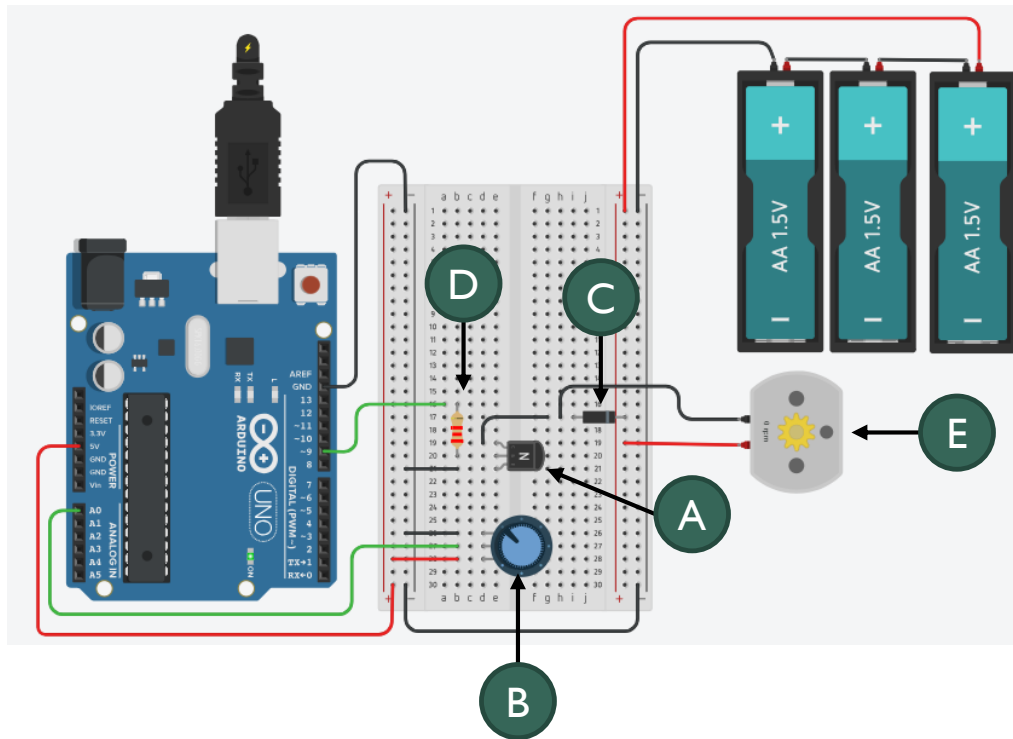
	Komponentit	Arduino pinnit
A	MOSFET (IRF520)	~9
B	Potentiometri	A0
C	Diodi	
D	DC-moottori	

Arduino

ArduinoCC

TinkerCAD

6. POTENTIOMETRILLÄ OHJATTU DC-MOOTTORI (TRANSISTORI)



	Komponentit	Arduino pinnit	Micro:bit pinnit
A	NPN Transistori (BC 337-25)	~9	P0
B	Potentio-metri	A0	P1
C	Diodi		
D	Vastus 2,2kΩ		
E	DC-moottori		

Arduino	Micro:bit
ArduinoCC	JSBlocks
TinkerCAD	mbed

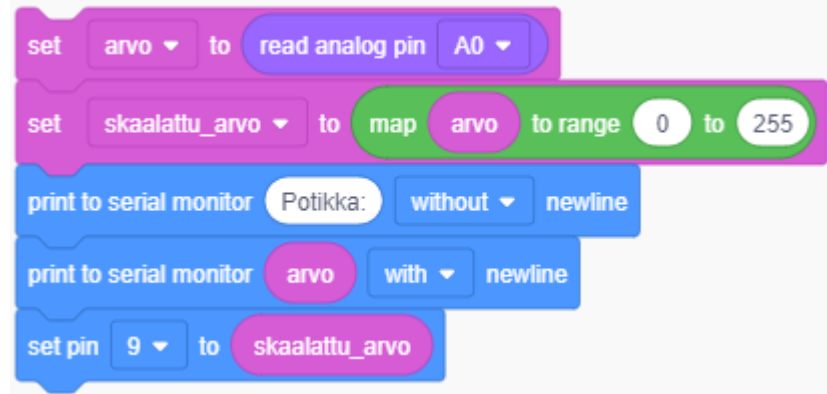
6. ARDUINO-OHJELMOINTI

AnalogRead, AnalogWrite ja map

analogRead
palauttaa arvon välillä
0-1023 mutta
analogWrite
kirjoittaa arvoja välillä
0-255, siksi luettu arvo
pitää skaalata käyttäen
map-funktiota. map-
funktiolle annetaan
lähtöalue (0-1023) ja
kohdealue (0-255).

```
int M_OHJAUS = 9;  
int POTIKKA = A0;  
int arvo = 0;  
int skaalattu_arvo = 0;
```

```
void setup() {  
    pinMode(M_OHJAUS, OUTPUT);  
    Serial.begin(9600);  
}  
  
void loop() {  
    arvo = analogRead(POTIKKA);  
    skaalattu_arvo = map(arvo, 0, 1023, 0, 255);  
    Serial.print("Potikan lukema: ");  
    Serial.println(arvo);  
    analogWrite(M_OHJAUS, skaalattu_arvo);  
    delay(10);  
}
```



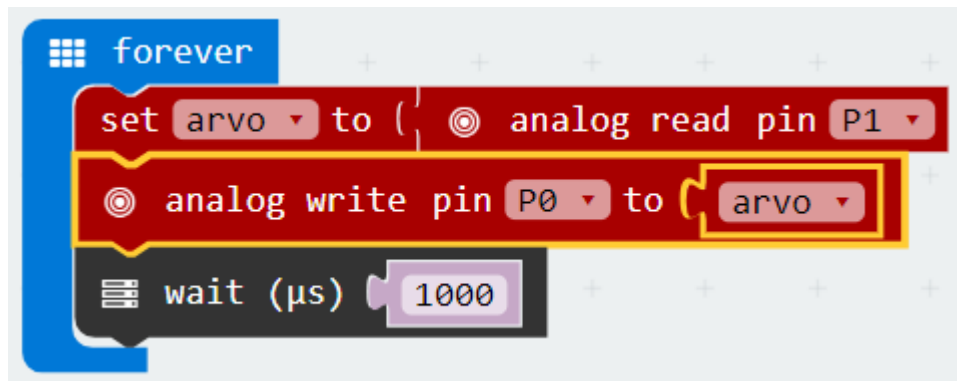
pinMode

pinMode:a ei tarvitse asettaa
pinneille A0-05, koska ne ovat
aina analogisia IN-pinnejä.
OUTPUT-pinnit on aina
alustettava setup-osiossa.

Eri kytkennät, sama koodi

Ohjelmoinnin näkökulmasta MOSFET ja transistori-
kytkennät eivät eroa toisistaan. Moottorin ohjauskoodi on
siis sama vaikka kytkennät ovatkin hieman erilaiset.

6. MICROBIT-OHJELMOINTI



```
#include "MicroBit.h"
```

```
MicroBit uBit;
```

```
int arvo = 0;
```

```
int main()
```

```
{
```

```
    uBit.init();
```

```
    while(1){
```

```
        arvo = uBit.io.P1.getAnalogValue();
```

```
        uBit.io.P0.setAnalogValue(arvo);
```

```
        uBit.sleep(1);
```

```
    }
```

```
}
```

getAnalogValue ja

setAnalogValue

Ohjelma on todellakin näin yksinkertainen: luetaan analogisen pinnin arvo (potentiometri) ja kirjoitetaan se toiseen pinniin (moottoria ohjaava transistori). Edes skaalausta ei tarvita, koska micro:bitillä sekä input että output arvot ovat välillä 0-1023.

7. LIIKEOHJATTAVA SOITIN

- Tässä työssä tutustutaan ultraäänisensorin käyttöön ohjausvälineinä. Sensorin havaitseman liikkeen avulla ohjataan pietsosummeria. Tutustu komponentteihin lukemalla niistä tietoiskut (ks. linkit).

- Tee kytkentä ohjeiden mukaisesti.

- Ohjelmointi:

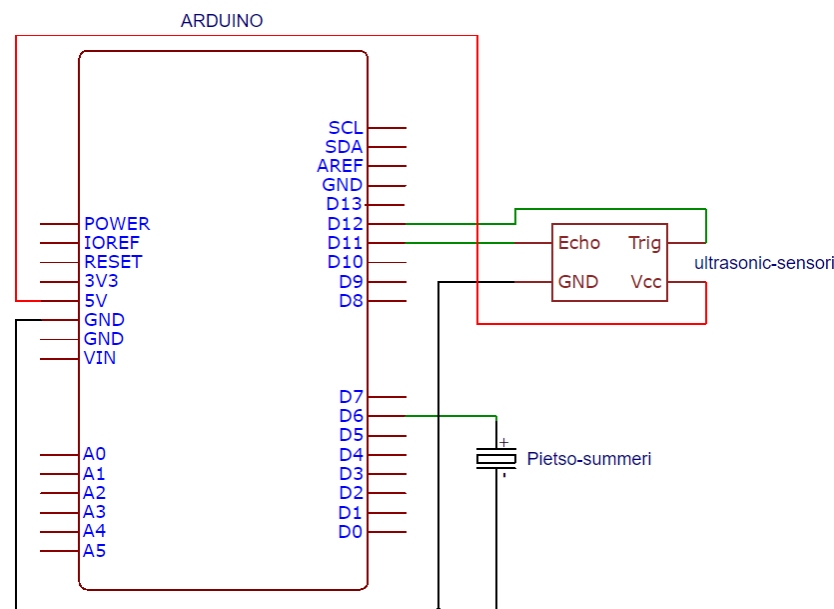
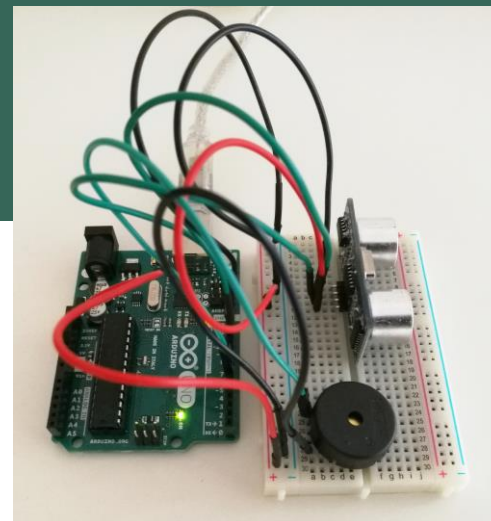
- Luetaan sensorin mittaama etäisyys (cm), päätellään mitä säveltä etäisyys vastaa (5cm per sävel) ja soitetään pietsosummerilla kyseistä äänentaajuutta (Hz).

- Tätä ohjelmaa varten joudut asentamaan ArduinoCC:hen kaksi kirjastoa (uusin zip-tiedosto):

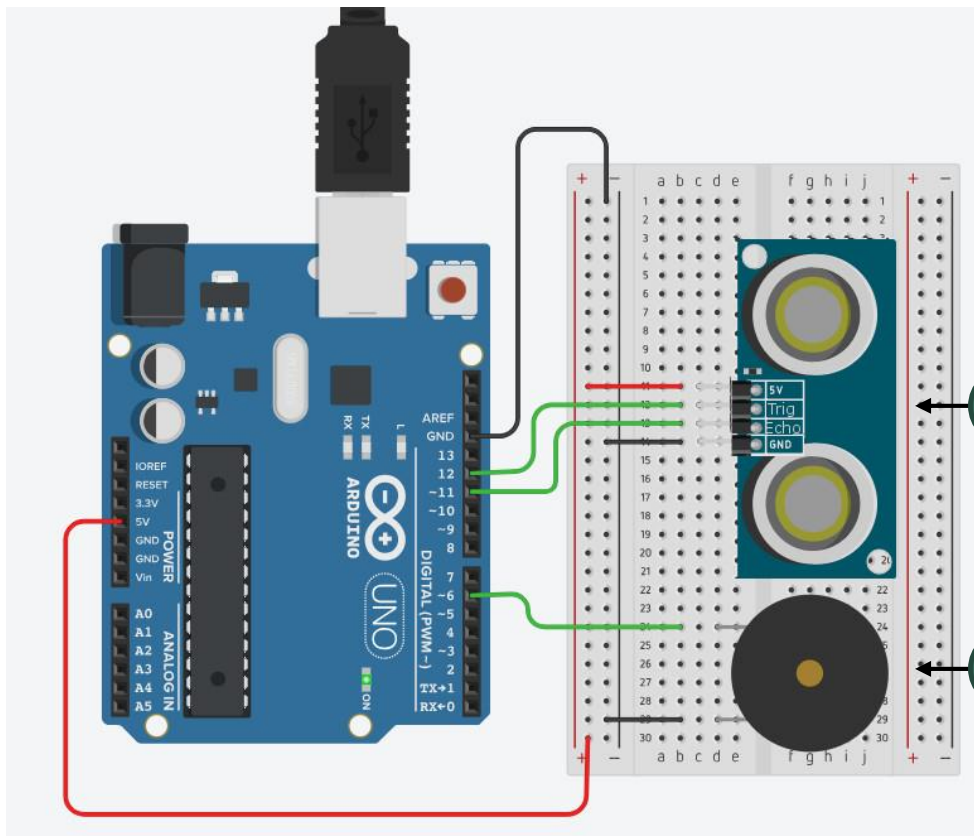
- <https://bitbucket.org/teckel12/arduino-new-ping/downloads/>

- <https://bitbucket.org/teckel12/arduino-new-tone/downloads/>

Katso kirjaston asennusohjeet täältä.



7. LIIKEOHJATTAVA SOITIN

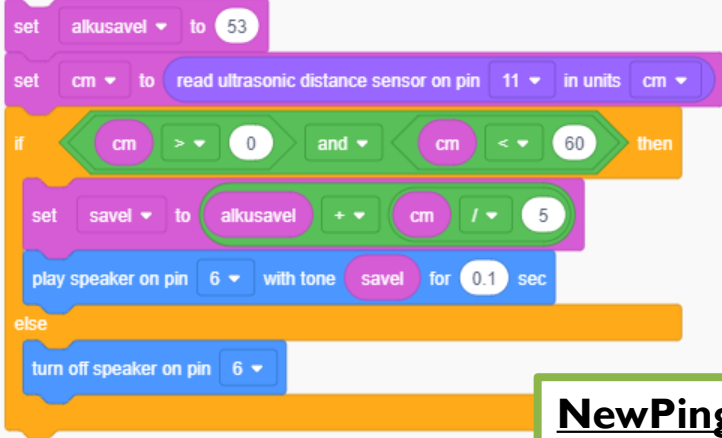


	Komponentit	Arduino pinnit	Micro:bit pinnit
A	Ultraääni-sensori HC-SR04 *)	Trig: 12 Echo: ~11	Trig: P2 Echo: P1
B	Pietso-summeri	~6	P0

*) joissakin ultraäänisensoreissa on vain yksi pinni (SIG), tällainen sensor on käytössä TinkerCAD-ohjelmassa (ks. taulukon linkki)

Arduino	Micro:bit
ArduinoCC	JSBlocks
TinkerCAD	

7. ARDUINO-OHJELMOINTI



```
#include <NewPing.h>
#include <NewTone.h>

NewPing sonar(12, 11, 200);
```

```
int etaisyyss = 0;
int askeleet = 0;
int taajuus = 0;
int MAX_cm = 60;
int ASKEL_cm = 5;
int SUMMERI = 6;

float c = 261.6, cis = 277.2, d = 293.7, dis = 311.1,
e = 329.6, f = 349.2, fis = 370.0, g = 392.0,
gis = 415.3, a = 440.0, ais = 466.2, h = 493.9,
c2 = 523.3, cis2 = 554.4, d2 = 587.3, dis2 = 622.3,
e2 = 659.3, f2 = 698.5, fis2 = 740.0, g2 = 784.0,
gis2 = 830.6, a2 = 880.0, ais2 = 932.3, h2 = 987.8;

float C_DUURI[] = {c, d, e, f, g, a, h, c2, d2, e2, f2, g2, a2, h2};
```

NewPing ja NewTone -kirjastot

Ulträänisensorin käyttö vaatii NewPing-kirjaston, ja koska se ei toimi tavallisen tone-funktion kanssa, otamme käyttöön myös NewTone-kirjaston, että äänet toimivat. NewPing – oliolle (sonar) annetaan alustuksessa Trig ja Echo pinnien numerot (12, 11) sekä maksimietäisyys (200 cm).

```
void setup() {
  pinMode(SUMMERI, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  etaisyyss = sonar.ping_cm();
  Serial.print("Etäisyys: ");
  Serial.print(etaisyyss);
  Serial.println(" cm");

  if(0 < etaisyyss && etaisyyss < MAX_cm){
    askeleet = etaisyyss / ASKEL_cm;
    taajuus = C_DUURI[askeleet];
    Serial.println(askeleet);
    Serial.println(taajuus);
    NewTone(SUMMERI, taajuus);
  } else {
    noNewTone(SUMMERI);
  }
  delay(100);
}
```

Taulukko []

Taulukko on tietorakenne, johon voi tallentaa tietoa. Tässä on tehty C_DUURI niminen taulukko johon on tallennettu kaksi oktaavia (14kpl) c-duurin säveliä (Hz). Alkioihin voi viitata niiden järjestysluvulla: C_DUURI[0] on c ja C_DUURI[7] on c2-sävel.

7. MICROBIT-OHJELMOINTI

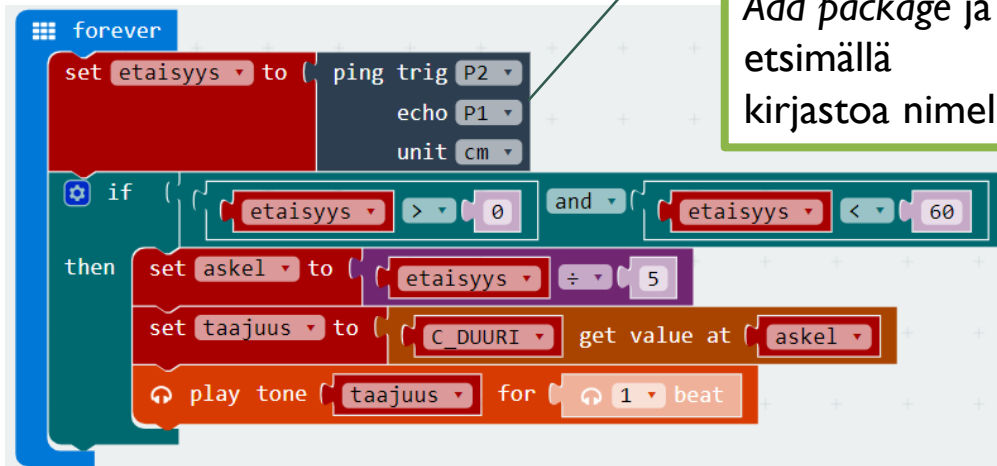


Taulukko (array)

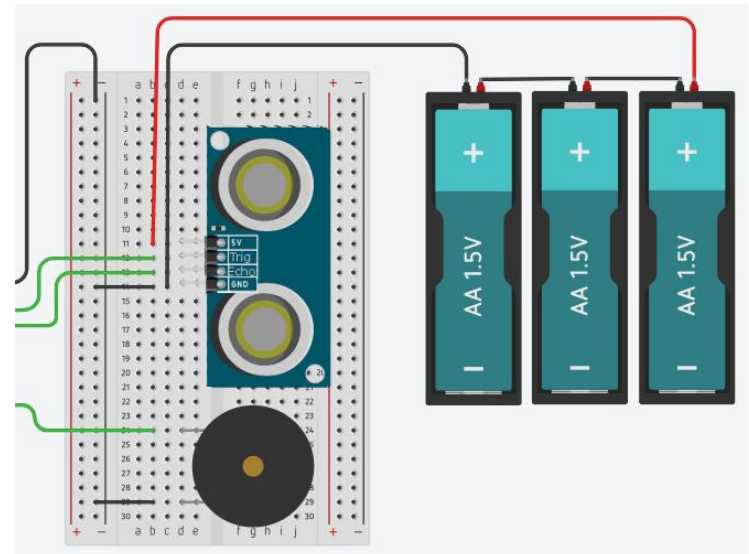
Taulukko on tietorakenne, johon voi tallentaa tietoa. Tässä on tehty C_DUURI niminen taulukko johon on tallennettu kaksi oktaavia (14kpl) c-duurin säveliä (Hz). Alkioihin voi viitata niiden järjestysluvulla: C_DUURI[0] on c ja C_DUURI[7] on c2-sävel.

Kirjasto (package)

Ultraäänisensorin käyttäminen vaatii lisäkirjaston (Sonar). Sen voi asentaa valikosta Add package ja etsimällä kirjastoa nimellä.



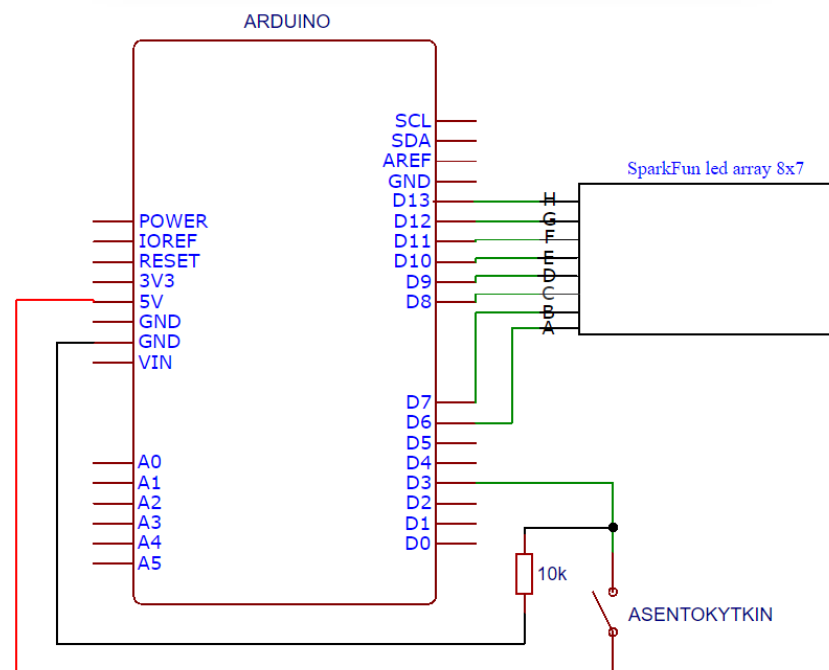
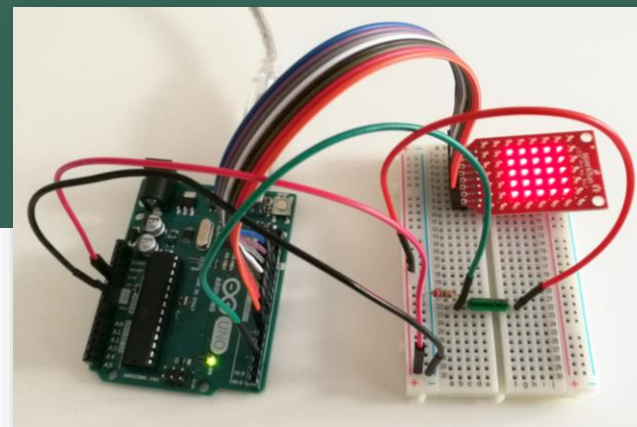
Ultraäänisensori vaatii 5V jännitteen, joten se ei toimi micro:bitin kanssa ilman ulkoista lisäparistoa (kolme 1,5V paristoa sarjaankytkettynä riittää). Huomaa, että paristojen negatiivinen napa pitää sensorin lisäksi kytkeä micro:bitin GND:hen.



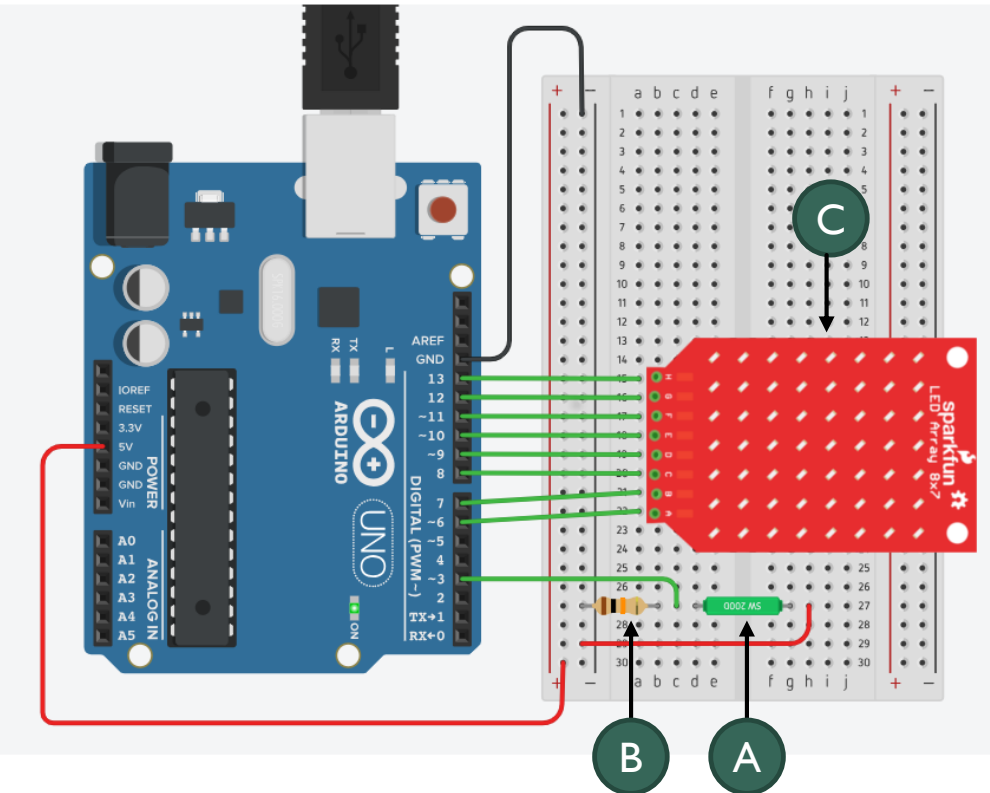
8. ASENNON TUNNISTIN

- Tässä työssä tutustutaan asentokytkimen sekä ledimatriisiin käyttöön. Tutustu komponentteihin lukemalla niistä tietoiskut (ks. linkit).
- Jos käytät Arduinoa tee kytkentä ohjeiden mukaisesti. Jos käytät micro:bit:iä et tarvitse ulkoista ledimatriisia. Huomaa, että SparkFun LED Array 8x7 vaatii kolvaamaan siihen pinnit.
- Ohjelmointi:
 - Kun asentokytkin on ”päällä”-asennossa, näytetään ledimatriisissa ”play”-kuviota (kolmio) ja kun se on ”pois päältä”-asennossa ”stop”-kuviota (neliö). Testaa ohjelmaa kääntämällä koekytkentälevyä eri asentoihin.
 - Tätä ohjelmaa varten joudut asentamaan ArduinoCC:hen kaksi kirjastoa:
 - <https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/yacll/Chaplex.zip>
 - https://github.com/sparkfun/SparkFun_LED_Array_8x7_Arduino_Library/archive/master.zip

Katso kirjaston asennusohjeet täältä.



8. ASENNON TUNNISTIN



	Komponentit	Arduino pinnit	Micro:bit pinnit
A	Asentokytkin	3	P0
B	Vastus 10kΩ		
C	Sparkfun	6	
	Led array	7	
	8x7	8	
		9	
		10	
		11	
		12	
		13	

Arduino	Micro:bit
<u>ArduinoCC</u>	<u>JSBlocks</u>
	<u>mbed</u>

8. ARDUINO-OHJELMOINTI

Kirjastot

SparkFun led array 8x7 vaatii kaksi kirjastoa
SparkFun.h ja Charplex.h.

```
#include <SparkFun_LED_8x7.h>
#include <Chaplex.h>
```

```
static byte LEDIPINNIT[] = {6, 7, 8, 9, 10, 11, 12, 13};
int ASENTO = 3;
int arvo = 0;
```

```
const byte PLAY[] = {0,1,0,0,0,0,0,0,
                    0,1,1,1,0,0,0,0,
                    0,1,1,1,1,1,0,0,
                    0,1,1,1,1,1,1,1,
                    0,1,1,1,1,1,0,0,
                    0,1,1,1,0,0,0,0,
                    0,1,0,0,0,0,0,0};
```

```
const byte STOP[] = {0,0,0,0,0,0,0,0,
                    0,1,1,1,1,1,0,0,
                    0,1,1,1,1,1,0,0,
                    0,1,1,1,1,1,0,0,
                    0,1,1,1,1,1,0,0,
                    0,1,1,1,1,1,0,0,
                    0,0,0,0,0,0,0,0};
```

Plex.init, Plex.display, Plex.drawBitmap

Kirjastoa käytetään Plex-olion kautta. Sille annetaan alustuksessa käytettyjen ledien pinnien numerot taulukkomuodossa. drawBitmap-metodilla luodaan piirrettävä kuva, joka saadaan näytöllä display-metodilla.

```
void setup() {
  Plex.init(LEDIPINNIT);
  Plex.clear();
  Plex.display();
  pinMode(ASENTO, INPUT);
  Serial.begin(9600);
}

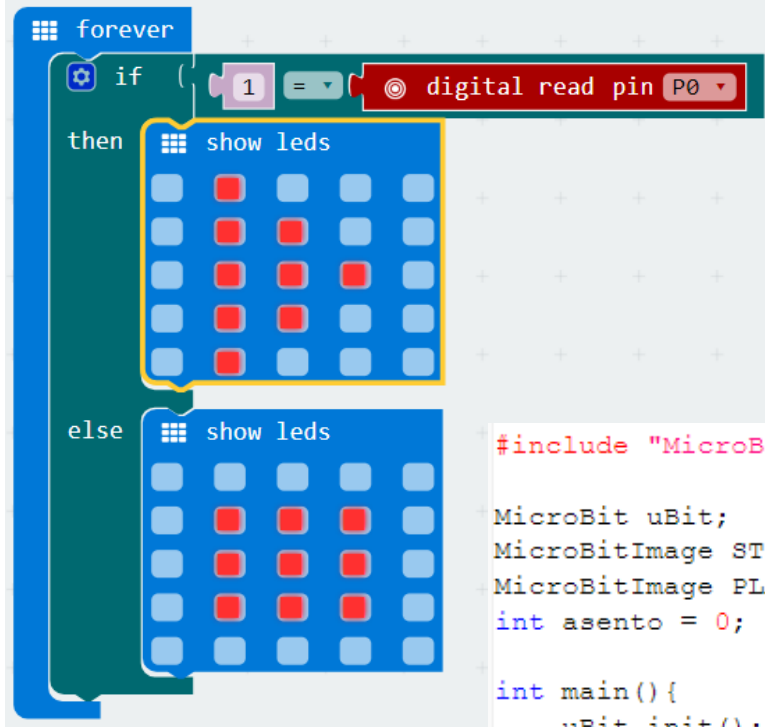
void loop() {
  arvo = digitalRead(ASENTO);
  Serial.println(arvo);
  if(arvo == HIGH) {
    Plex.drawBitmap(PLAY);
    Plex.display();
    delay(1000);
  } else {
    Plex.drawBitmap(STOP);
    Plex.display();
    delay(1000);
  }
}
```

BitMaps

Kuviot tallennetaan bitmap-muodossa eli taulukkona, jossa ilmoitetaan ledin tila (1=päällä, 0=pois). byte tarkoittaa 8 bittistä esitysmuotoa ja const varmistaa sen, että muuttujien PLAY ja STOP arvoja ei muuteta ohjelman suorituksen aikana.

static määrää sen, että muuttuja luodaan ohjelman alussa ja tuhotaan vasta sen lopussa.

8. MICROBIT-OHJELMOINTI



```
#include "MicroBit.h"

MicroBit uBit;
MicroBitImage STOP("255,255,255\n255,255,255\n255,255,255\n");
MicroBitImage PLAY("255,0,0\n255,255,0\n255,255,255\n255,255,0\n255,0,0\n");
int asento = 0;

int main(){
    uBit.init();
    while (1){
        asento = uBit.io.P0.getDigitalValue();
        MicroBitImage image(5,5);
        if(asento == 1){
            image.paste(PLAY, 1, 0);
            uBit.display.print(image);
            uBit.sleep(1000);
        } else {
            image.paste(STOP, 1, 1);
            uBit.display.print(image);
            uBit.sleep(1000);
        }
    }
}
```

MicroBitImage

Kuviot tallennetaan `MicroBitImage` –olioina. Ledien tilat ilmoitetaan riveittäin (rivin loppuun merkitään rivin vaihto `\n`) ja ledien arvot erotetaan pilkulla (255=kokonaan päällä, 0=pois). Tässä on määritelty vain ne ruudun osat joissa on ledi päällä (3x3 ja 3x4 kokoiset kuvat). Nämä pienemmät kuvat liitetään myöhemmin isompaan 5x5 pohjaan.

MicroBitImage

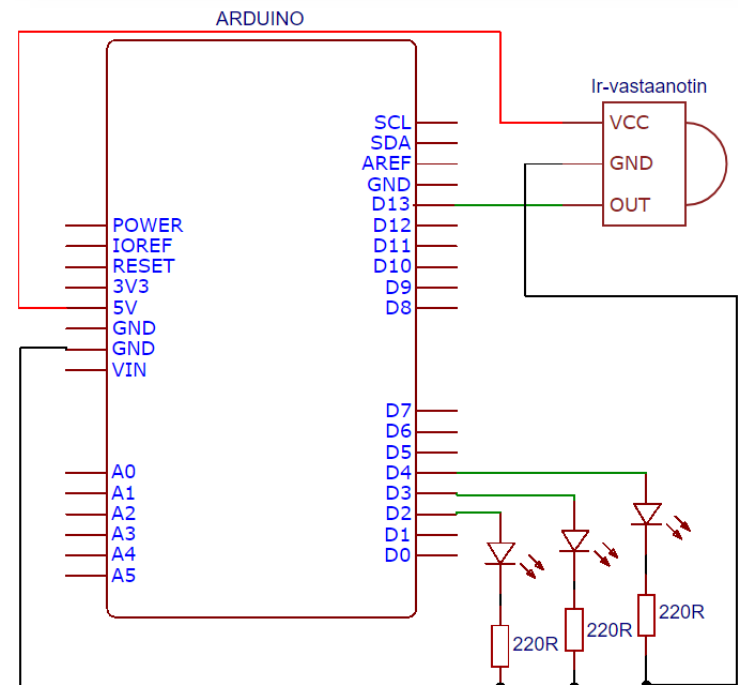
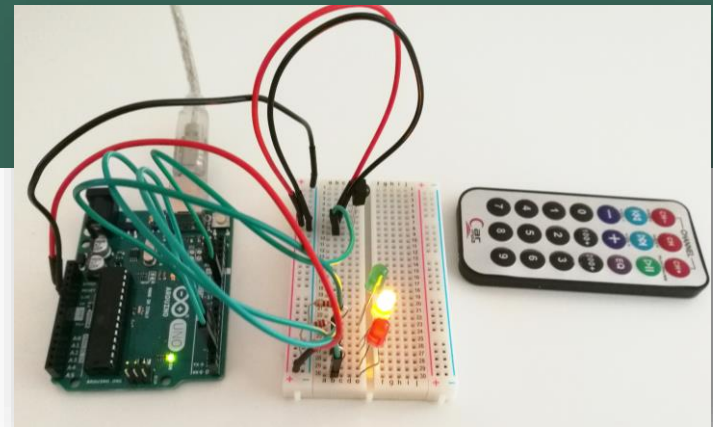
Tässä luodaan tyhjä 5x5 kuva, jonka päälle pienemmät kuvat liitetään paste-metodilla. Paste saa parametrina kuvan, sekä x- ja y-koordinaatit johon se sijoitetaan. Vasen yläkulma on 0, 0.

uBit.display.print

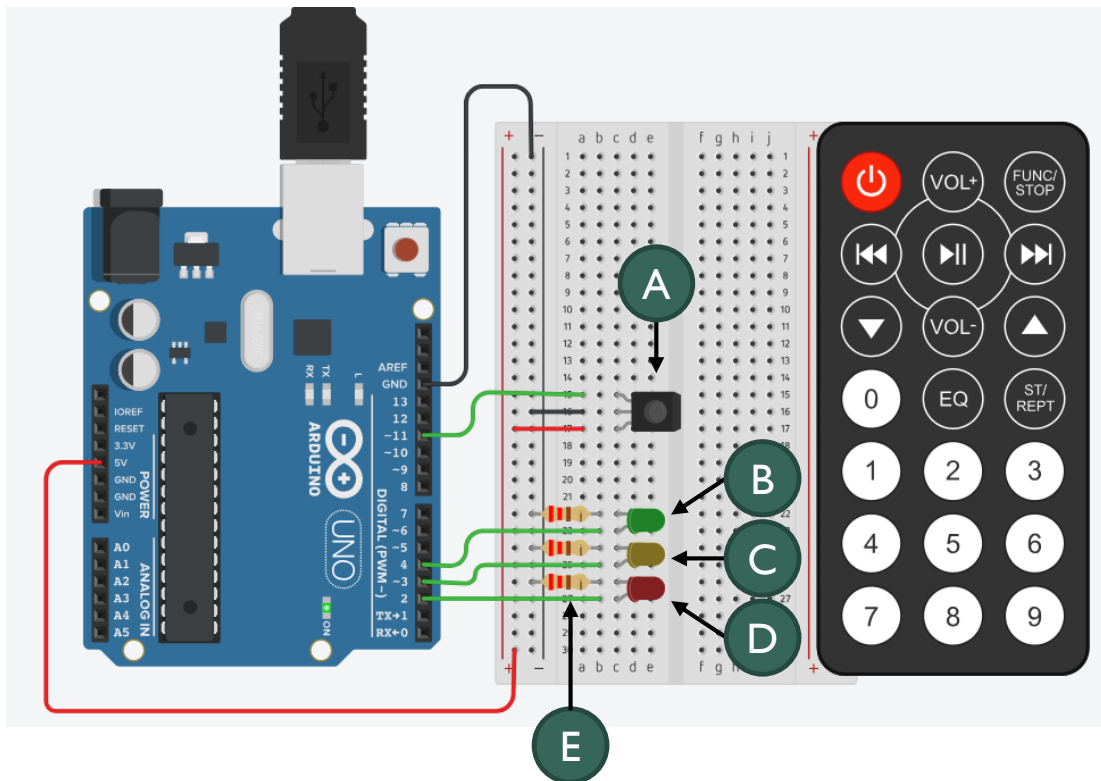
Kuviot saadaan näytölle käyttämällä `uBit.display.print`-metodia.

9. KAUKO-OHJATTAVAT LEDIVALOT

- Tässä työssä tutustutaan IR-lähettimen ja -vastaanottimen toimintaan. Kaukosäätimellä ohjataan ledejä. Tutustu komponentteihin lukemalla niistä tietoiskut (ks. linkit).
- Tee kytkentä ohjeiden mukaisesti.
- Ohjelmointi:
 - Luetaan IR-vastaanottimelta saatu viesti. Jos viesti kertoo, että näppäintä 1 on painettu, sytytetään punainen ledi, näppäimestä 2 sytytetään keltainen ledi ja näppäimestä 3 vihreä ledi. Jos kyseinen ledi on jo päällä kun näppäintä painetaan, se sammutetaan.
 - Asenna ArduinoCC:hen tämä kirjasto:
 - <http://z3t0.github.io/Arduino-IRremote/>



9. KAUKO-OHJATTAVAT LEDIVALOT



	Komponentit	Arduino pinnit
A	IR-vastaanotin	~11
B	Vihreä ledi	4
C	Keltainen ledi	3
D	Punainen ledi	2
E	3x vastus 220Ω	

Arduino

ArduinoCC

TinkerCAD
(c-koodia)

9. ARDUINO-OHJELMOINTI

```
#include <IRremote.h>
#include <IRremoteInt.h>
```

```
int PUN = 2;
int KELT = 3;
int VIHR = 4;
```

```
int VASTAANOTIN = 11;
IRrecv ir(VASTAANOTIN);
decode_results viesti;
```

```
void setup() {
  Serial.begin(9600);
  ir.enableIRIn();
  pinMode(PUN, OUTPUT);
  pinMode(KELT, OUTPUT);
  pinMode(VIHR, OUTPUT);
}
```

```
bool punPalaa = false;
bool keltPalaa = false;
bool vihrPalaa = false;
```

Kirjastot

IR-lähettimen ja – vastaanottimen käyttö vaatii IRremote.h ja IRremoteInt.h kirjastot ja yhden IRrecv-olion (ir).

Totuusarvot (true/false)

Tässä tehdään muuttujat, joiden tyyppi on totuusarvo (bool), eli ne voivat saada vain kaksi arvoa, tosi (true) tai epätosi (false).

```
void loop() {
  if(ir.decode(&viesti)) {
    Serial.println(viesti.value, HEX);
    ir.resume();
  }
  if(ir.decode(&viesti)) {
    if (viesti.value == 0xFF30CF) {
      if (punPalaa == true) {
        digitalWrite(PUN, LOW);
        punPalaa = false;
      } else {
        digitalWrite(PUN, HIGH);
        punPalaa = true;
      }
    }
    if (viesti.value == 0xFF18E7) {
      if (keltPalaa == true) {
        digitalWrite(KELT, LOW);
        keltPalaa = false;
      } else {
        digitalWrite(KELT, HIGH);
        keltPalaa = true;
      }
    }
    if (viesti.value == 0xFF7A85) {
      if (vihrPalaa == true) {
        digitalWrite(VIHR, LOW);
        vihrPalaa = false;
      } else {
        digitalWrite(VIHR, HIGH);
        vihrPalaa = true;
      }
    }
  }
  ir.resume();
}
```

Referenssi (&)

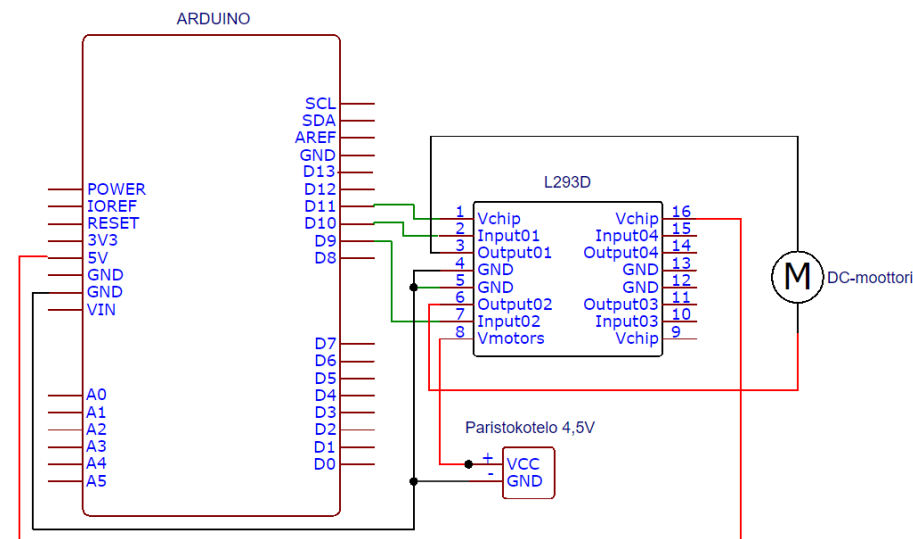
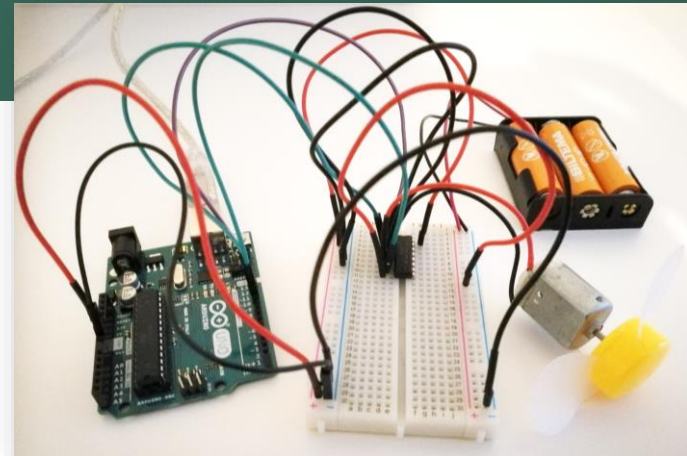
Joskus funktiolle annetaan muuttujan arvon sijasta sen osoite eli referenssi/osoite (muuttujan nimen eteen kirjoitetaan & merkki).

Heksadesimaaliluvut

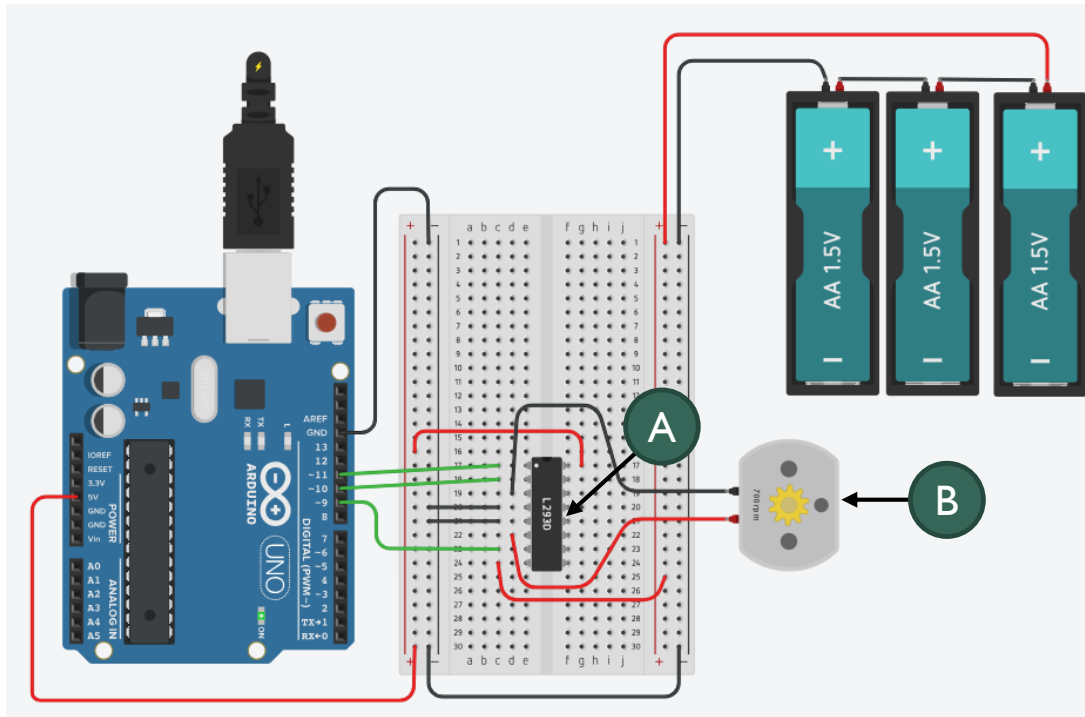
IR-lähetin lähettää tietyn heksadesimaaliluvun (alkaa 0x), kun jotain sen näppäintä painetaan. Saat ne selville sarjamonitorin avulla.

I0. MOOTTORIN OHJAAMINEN H-SILLAN AVULLA

- Tässä työssä tutustutaan siihen miten H-sillan avulla ohjataan tasavirta-moottoria. Tutustu komponentteihin lukemalla niistä tietoiskut (ks. linkit).
- Tee kytkentä ohjeiden mukaisesti Arduinolle. Micro:bit vaatii toisen setin lisäparistoja, koska H-sillan tarvitsee 5V jännitteen toimiakseen (kytkentäkuva micro:bit - ohjelmointiohjeiden yhteydessä).
- Ohjelmointi:
 - Ohjataan moottori kiihdyttämään nostamalla nopeuspinnin jännitettä silmukan sisällä. Kun maksiminopeus on saavutettu, pysäytetään moottori ja käännetään se pyörimään takaisin päin ja samalla hidastetaan pyörimisnopeutta,



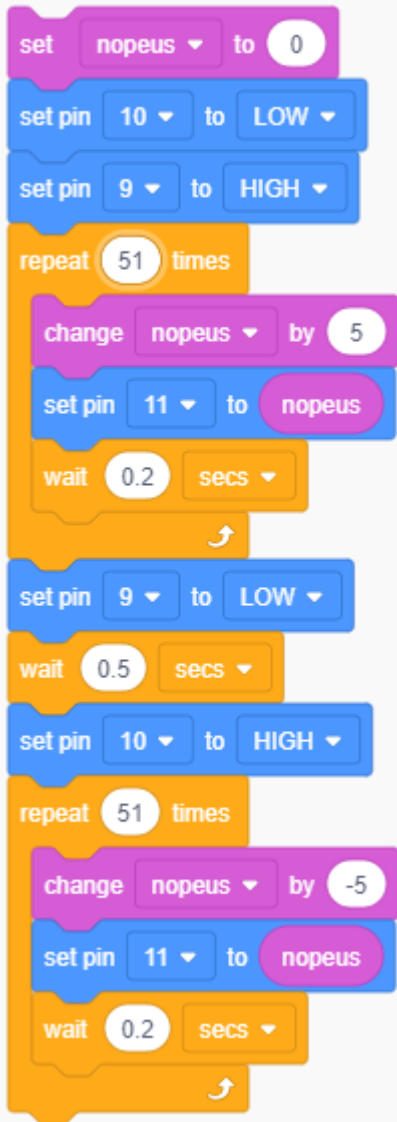
I0. MOOTTORIN OHJAAMINEN H-SILLAN AVULLA



	Komponentit	Arduino pinnit	Micro:bit pinnit
A	H-silta (L293D)	~11	P2
		10	P1
		9	P0
B	DC-moottori		

Arduino	Micro:bit
ArduinoCC	JSBlocks
TinkerCAD	mbed

10. ARDUINO-OHJELMOINTI



```
#define NOPEUS 11
#define OHJAUS1 10
#define OHJAUS2 9

void setup() {
  pinMode(NOPEUS, OUTPUT);
  pinMode(OHJAUS1, OUTPUT);
  pinMode(OHJAUS2, OUTPUT);
}

void loop() {
  digitalWrite(OHJAUS1, HIGH);
  digitalWrite(OHJAUS2, LOW);
  for(int i=0; i<=255; i=i+5){
    analogWrite(NOPEUS, i);
    delay(200);
  }
  digitalWrite(OHJAUS1, LOW);
  delay(500);
  digitalWrite(OHJAUS2, HIGH);
  for(int i=255; i>=0; i=i-5){
    analogWrite(NOPEUS, i);
    delay(200);
  }
  digitalWrite(OHJAUS2, LOW);
  delay(500);
}
```

#define

Aikaisemmin olemme käyttäneet muuttujia pinnien numeroiden tallentamiseen. Toinen vaihtoehto on käyttää `define`:ä, joka etsii koodista kohdat joissa lukee `NOPEUS`, ja korvaa ne annetulla arvolla 11 ennen ohjelman suorittamista. Tätä toiminnallisuutta kutsutaan makroksi.

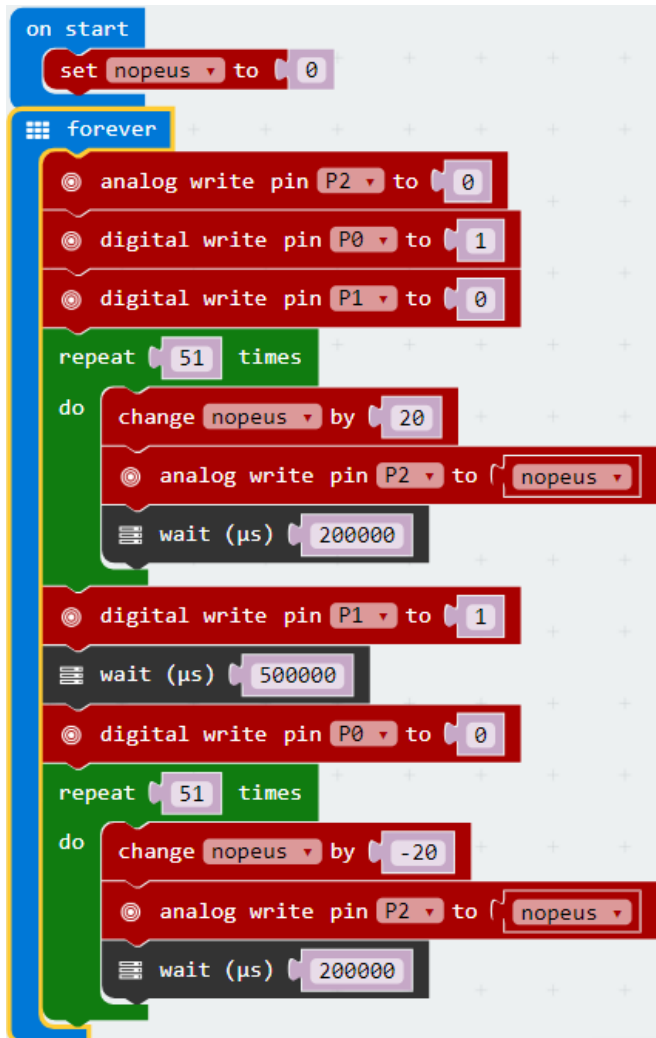
For-silmukka

Tässä `for`-silmukassa on sisäinen muuttuja `i`, joka kasvaa nolasta 255:een viiden askelin. Tämä arvo annetaan moottorille, joka alkaa kiihdyttämään.

Moottorin ohjaus

Moottorin ohjaus tapahtuu kolmen pinnin avulla. Kun `OHJAUS1` ja `OHJAUS2` ovat erit ja `NOPEUS` pinni ei ole nolla, moottori pyörii. Tässä kohtaa moottori on pysähtynyt.

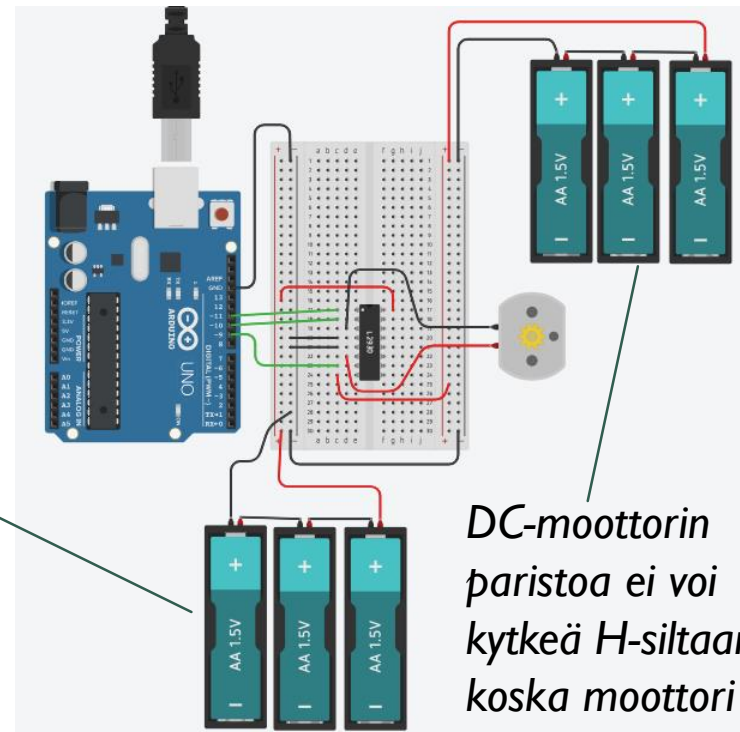
10. MICROBIT-OHJELMOINTI



H-silta vaatii 5V jännitteen, joten se ei toimi micro:bitin kanssa ilman ulkoista paristoa.

```
MicroBit uBit;

int main(){
  uBit.init();
  while (1){
    uBit.io.P1.setDigitalValue(1);
    uBit.io.P0.setDigitalValue(0);
    for(int i=0; i<=1023; i=i+20){
      uBit.io.P2.setAnalogValue(i);
      uBit.sleep(200);
    }
    uBit.io.P1.setDigitalValue(0);
    uBit.sleep(500);
    uBit.io.P0.setDigitalValue(1);
    for(int i=1023; i>=0; i=i-20){
      uBit.io.P2.setAnalogValue(i);
      uBit.sleep(200);
    }
    uBit.io.P0.setDigitalValue(0);
    uBit.sleep(500);
  }
}
```



DC-moottorin paristoa ei voi kytkeä H-siltaan, koska moottori aiheuttaa häiriötä H-sillan toimintaan.

Moottorin ohjaus

Moottorin ohjaus tapahtuu kolmen pinnin avulla. Kun P1 ja P0 ovat erit ja P2 ei ole nolla, moottori pyörii. Tässä kohtaa moottori on pysähtynyt.

KIRJASTON ASENTAMINEN

Uuden laitteen käyttö vaatii usein kirjaston asentamisen.

ArduinoCC

- Kirjasto ladataan netistä valmistajan sivustolta .zip – pakettina.
- Zip-paketti tuodaan ArduinoCC:hen valitsemalla valikosta:

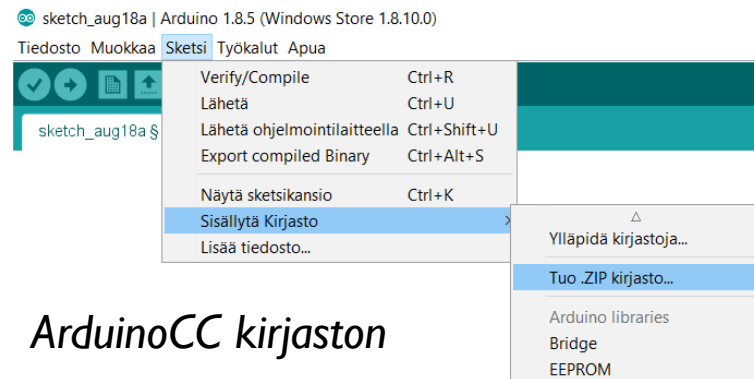
Sketsi → Sisällytä Kirjasto → Tuo .ZIP-kirjasto

JSBlocks

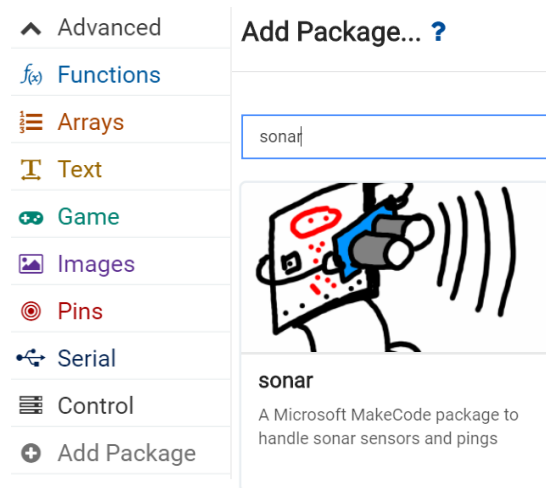
- Valitse Advanced-valikosta *Add package*
- Etsi nimellä ja valitse kirjasto hiirellä

mbed

- Valitse valikosta *Import*, etsi nimellä ja tuplaklikkaa kirjastoa



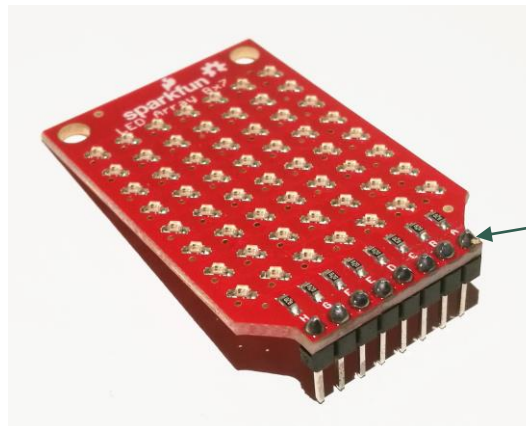
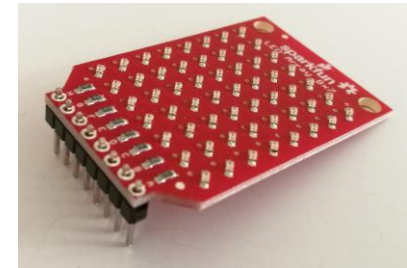
ArduinoCC kirjaston asennus



JSBlocks kirjaston asennus

SPARKFUN LED ARRAY 8X7 PINNIRIMAN JUOTTAMINEN

- SparkFun Led Array 8x7 toimitetaan ilman pinnejä. Jotta sen saa kiinnitettyä koekytkentälevyyn, levyn reikiin laitetaan pinnirimasta pinnit, jotka juotetaan kiinni. Huom! Juotokset vaativat tarkkuutta!



juotokset