

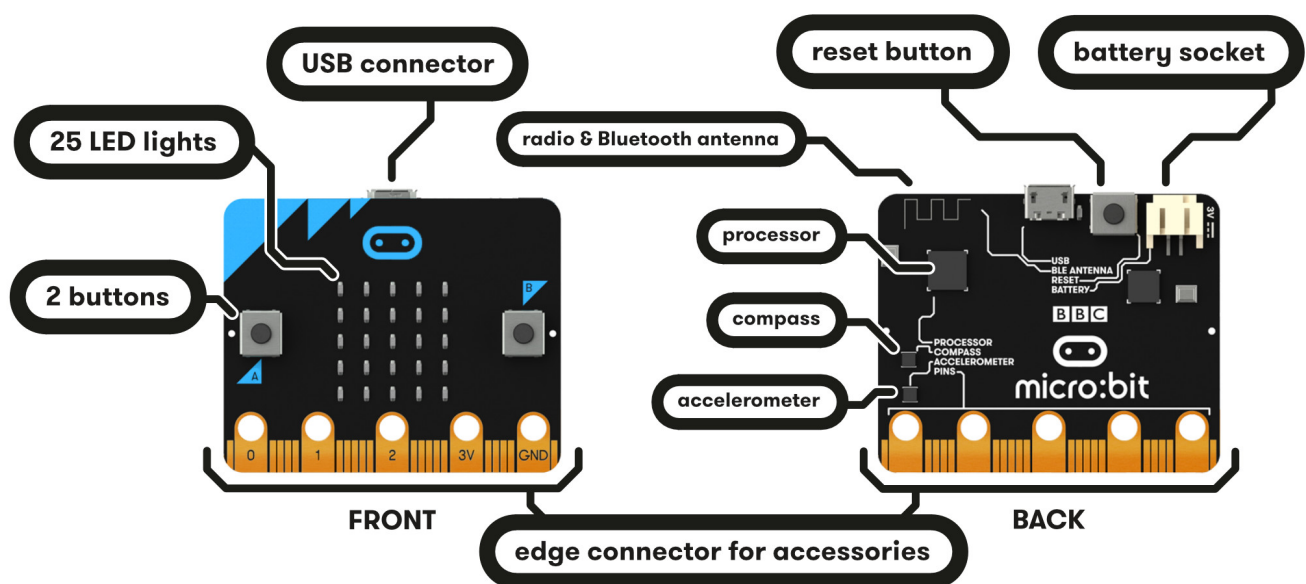
BBC Micro Bit ehk micro:bit

Tegemist on BBC (*British Broadcasting Corporation*) poolt arendatud **sardsüsteemiga** (*ingl embedded system*), mis on sobilik kasutamiseks hariduses. Tema iseloomustamiseks kasutatakse ka mõistet **avatud lähtekoodiga riistvara** (*ingl open source hardware*). Microbit kasutab ARM arhitektuuriga protsessorit, mis omakorda kuulub nn RISC (*ingl reduced instruction set computer*) protsessorite perekonda (kärbitud käsustikuga protsessor). Leidsin veel ühe „nunnu” sõna selle kohta: kärbikprotsessor.

Micro:biti ülesehitus

Microbiti mõõtmed on 4 x 5 cm ja tal on küljes mitmed huvitavaid kasutus- ja katsetusvõimalusi pakkuvad osad (Joonis 1).

Praeguseks on lisandunud microbiti versioon 2, kus on ühelt poolt parandatud olemasolevaid vidinaid ja teisalt on näiteks lisandunud kõlar (*speaker*) ja mikrofoni. Meil on kasutada 1. versioon. Lisa saab 2. versiooni kohta uurida microbiti kodulehelt (<https://microbit.org/get-started/user-guide/overview/>)



Joonis 1: Micro:bit ja tema osad (Allikas: <https://microbit.org/guide/features/>)

Microbiti programmeerimiskeskonnad

Võimalusi microbiti programmeerimiseks on üldiselt kaks:

- lohistada kokku plokkide (sarnaselt Scratchile);
- kirjutada kood programmeerimiskeeles, millel on olemas microbiti tugi (näiteks Python, JavaScript).

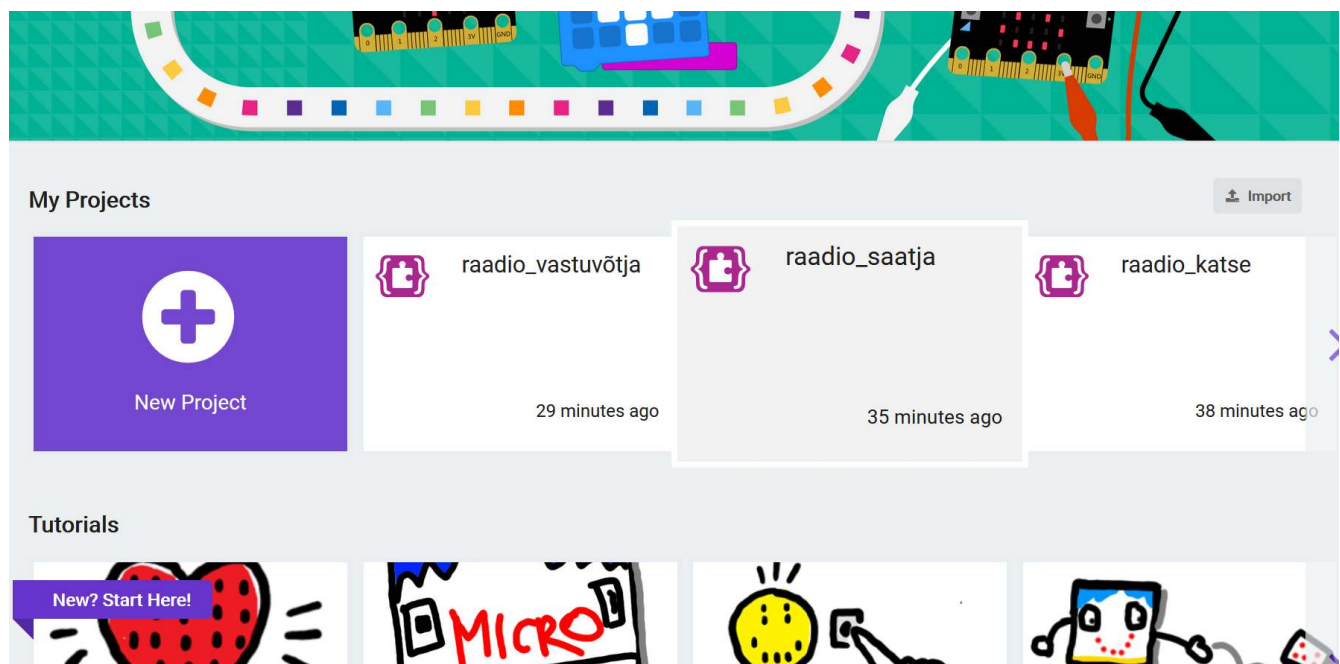
Plokkide abil programmeerimiseks on peamine soovitatav töökeskkond **Makecode**, kus plokkide ülesehitus on Scratchile sarnane: <https://makecode.microbit.org/>. Alguses olid kasutuses veidi teistsuguse kujuga (puzzle moodi) plokid ja nende kohta on internetis palju õpetusi. Vanemad plokiredaktori lingid toovad praegu Makecode lehele. Aga ka näiteks Scratchile on võimalik lisada microbiti plokid.

Programmeerimiskeeltest näivad eelistatud olevat Python ja JavaScript. Microbiti leheküljel suunab kasutama Pythonit ja MakeCode keskkonnas saab plokkide „tõlkida” JavaScripti. Väidetavalt on olemas ka võimalused teiste keelte jaoks. Kui programmeerimiskeelega Python on sõbralikud suhted olemas, võib katsetada Pythoni kogukonna poolt loodud ja ülalhoitavat keskkonda <https://python.microbit.org/v/2.0> või siis selle uuemat versiooni <https://python.microbit.org/v/beta> Et microbitiga suhtlemiseks vajalikud funktsioonid kätte saada, tuleb oma programmis kindlasti importida moodul microbit: `from microbit import *`

Microbiti osad ja näiteid nende programmeerimisvõimalustest

Järgneva aluseks on materjal: <https://microbit.org/guide/features/>. Kirjeldatakse lühidalt microbiti vidinaid ning võimalusi nende kasutamiseks programmeerimisel. Programmeerimise näited on tehtud kasutades plokkprogrammeerimise keskkonda MakeCode. Keskkond jääb siin materjalis pikemalt tutvustamata, sest varasem tutvus Scratchiga peaks andma piisavalt eelteadmisi. Jälgides näidiskoodis olevate plokkide värve, on need keskkonnas üsna lihtsasti leitavad.

Siiski paar vihjet keskkonna kohta annaksin.

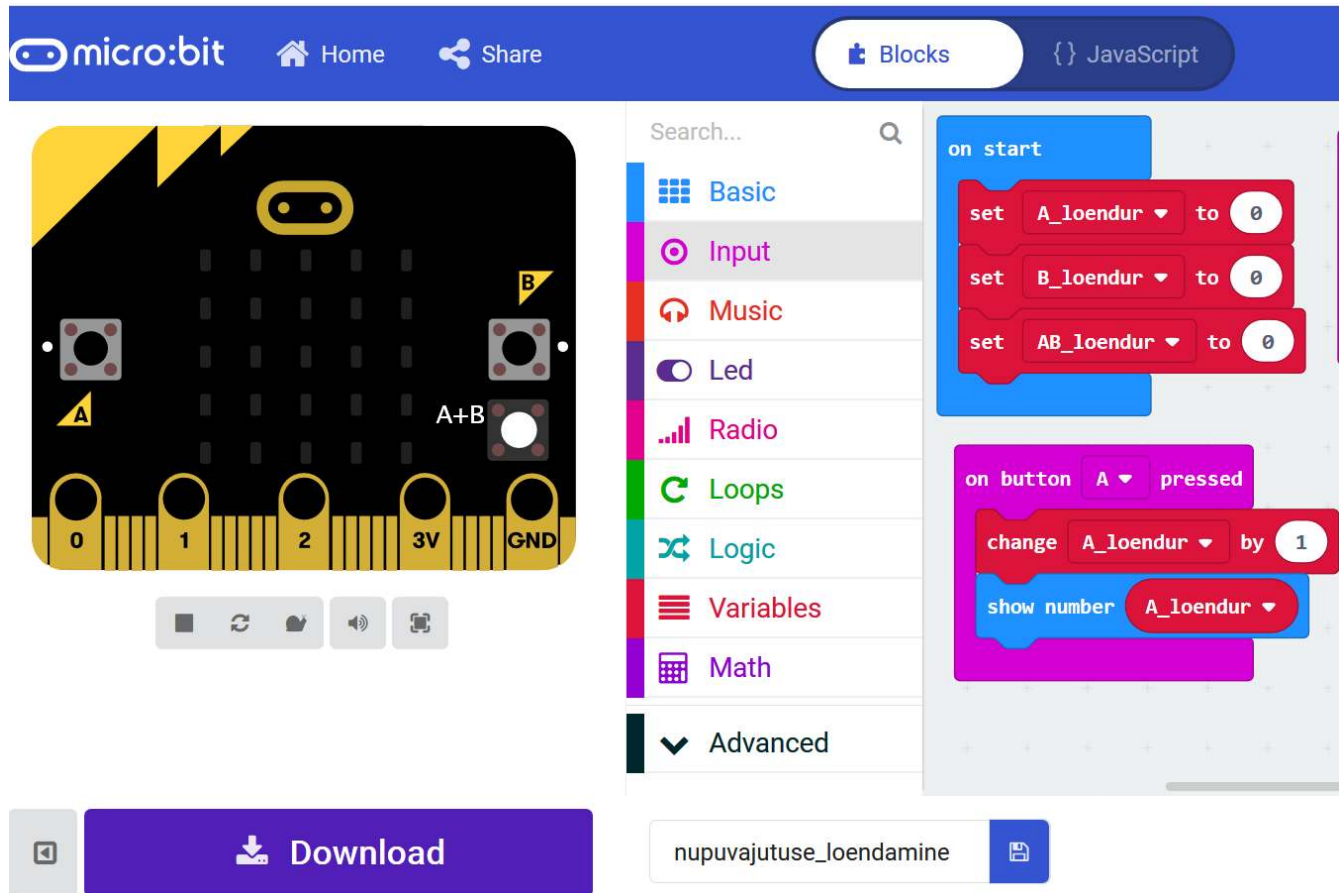


Joonis 2: Keskkonna MakeCode.microbit.org avapilt

Avalehelt (vt Joonis 2) saab alustada uue projekti tegemist. Samas on leitavad ka osa Sinu tehtud projekte. Pead arvestama sellega, et nende näitamine on arvutipõhine. Projektid salvestatakse lokaalselt

arvutisse ja sealt tuleb nad endaga teise arvutisse kaasa võtta. Tunni lõpus kopeeri salvestatud failid endale ära (kui tahad neid edaspidi kasutada).

Kui keskkond ei näita soovitud faili, saad selle **import**-nupu abil kohalikust arvutist avada. Tillukese nupu leiad sama ekraani parempoolsest servast, projektide nimekirja kohalt.



Joonis 3: MakeCode koodikuva.

Kui oled programmi valmis saanud ja töökeskkonnas simulaatoris ära testinud, on õige aeg ka tegeliku microbitiga katsetada. Selleks tuleb kood salvestada ja microbitti laadida. Selle töö tegemiseks pakutakse erinevaid variante. Mulle tundus kõige loogilisem Save-nupu abil (koodiplokkide all, Download nupust paremal (vt Joonis 3)) programm salvestada kohalikku arvutisse ja seejärel kopeerida fail arvuti külge ühendatud microbitti. Seadme ühendamisel ilmub File Exploreris (sarnaselt nt mälupulgale) uus seade, millel nimeks MICROBIT.

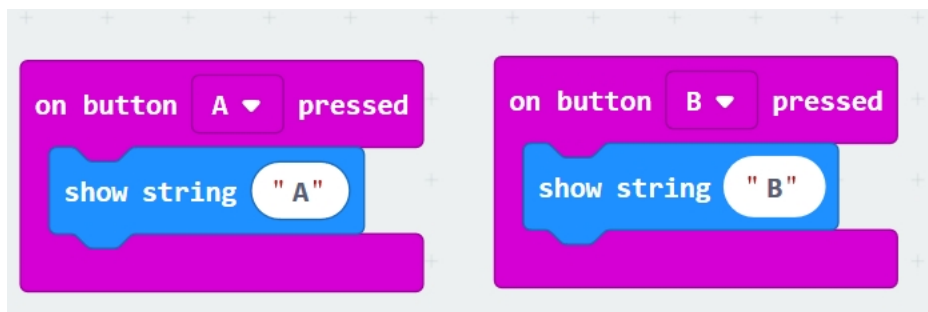
Nupud A ja B

Microbiti esipoolel on kaks nuppu, mis on tähistatud tähtedega A ja B. Nuppude vajutamisel (A, B või mõlemad korraga - A+B) tekib **sündmus** (*ingl event*) ning selle toimumisel on võimalik käivitada kood. Sellist sündmustega seotud programmikoodi käivitamist nimetatakse **sündmusepõhiseks programmeerimiseks** (*ingl event-driven programming*). Idee peaks olema tuttav Scratchist (rohelisel lipukesel klõpsides hakkab kood tööle, nooleklahvi vajutuse peale sprait liigub).

Näiteks võib nupu vajutamise siduda pildi / teksti muutusega LED paneelil, kasutaja / mängija-poolsete jah/ei valikutega ja miks mitte üldse kogu programmi käivitamisega. Üldisemalt: ükskõik millise tegevuse käivitamisega.

Ülesanne 1

Teeme nii, et nupu A vajutamisel tekib LED-paneelile A täht ja B vajutamisel B täht (Joonis 4). Ning mõlema nupu vajutamisel AB.



Joonis 4: Nupude A ja B vajutamise kood

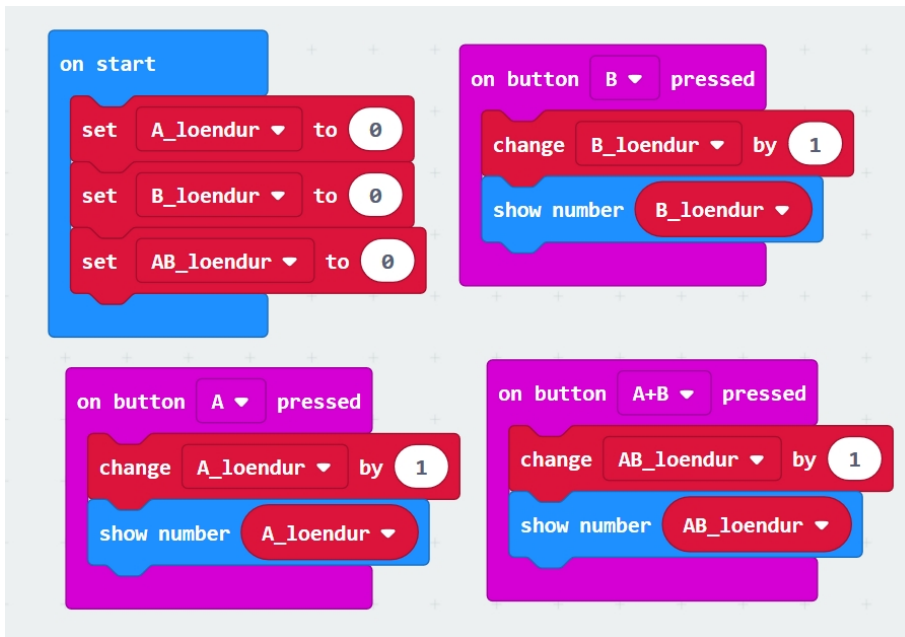
Ülesanne 2

Loendame nuppude vajutusi: mitu korda on vajutatud nuppu A, mitu korda nuppu B. Selleks vajame kolme **muutujat**. Muutujad (ingl *variable*) teeme valmis sarnaselt Scratchile. Nuppude vajutamise sündmustega seome vastavate muutujate suurendamise ühe võrra ja muutuja väärtuse näitamise (vt Joonis 5). Nagu keskkonnas näha, saab siin muutujale väärtust omistada ja/või seda muuta. Kõik rehkendamise plokid leiad sektsioonist **Math**.

Sündmuse plokk **on start** kirjeldab seda, mida tehakse programm käivitumisel. Sinna saab lisada käsud, mida tuleks teha enne, kui nuppudele vajutama hakkame. Antud juhul on selles kohas loendamiseks kasutatavate muutujate algväärtustamine nulliga (ehkki nulliga algväärtustamine peaks toimuma ka automaatselt).

Kohe alguses on ekraanil ka tsükli plokk **forever**. See plokk võimaldab hoida töös midagi, mis kulgeb taustal - näiteks muusika. Plokk ei sega microbitil teiste sündmuste püüdmist ja nende järgi töötamist.

Millal programm käivitub ehk millal tekib sündmus *on start*? Koodimise keskkonnas siis, kui kood kokku saab. Microbitis kohe peale koodi laadimist. Mõnikord tekib aga küsimus, kuidas koodi peatada. Selleks on koodimiskeskkonnas kohe microbiti kujutise all mõned hallid nupud (vt Joonis 3). Microbitil endal on tagaküljel ülal servas lähtestamise nupp (vt Joonis 1, *reset button*). Koodi peatamise soov tekib inimesel tavaliselt siis, kui õnnestub käima panna mingi tsükkel, mis ise lõppeda ei taha.



Joonis 5: Nupuvajutuste loendamise kood

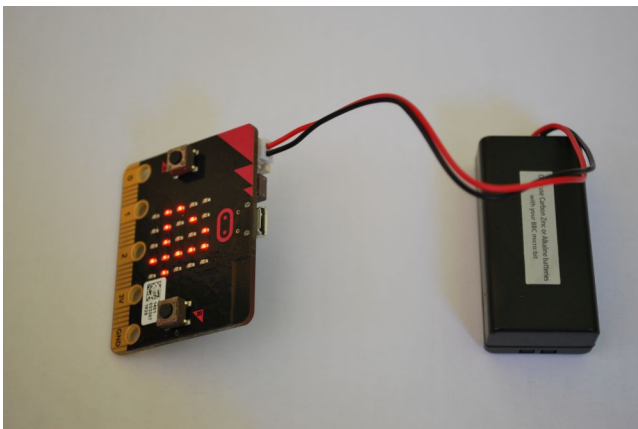
Veel ülesandeid:

1. Muus osas sama, mis eelmine ülesanne. Aga mõlema nupu vajutamine annab teada, kui palju on kokku nuppu A ja nuppu B vajutatud.
2. Genereeri nuppu A vajutades üks juhuslik arv, nuppu B vajutades teine juhuslik arv (otsusta ise vahemiku üle). Nuppe A ja B korraga vajutades too ekraanile arvude korrutis.

Kõigepealt saame oma koodi tööd kontrollida microbiti simulaatoris, mille leiad MakeCode keskkonna vasakus servas. Aga palju huvitavam küsimus on, kuidas saada kood tegelikult microbitti?

Programmi laadimine microbitti

Programmi laadimiseks microbitti on vaja lisaks USB kaablit. Et microbit töötaks, on tal vaja ka voolu. Kui microbit on USB kaabliga arvutiga ühendatud, saab ta arvutist toidet. Tee järgmised sammud:



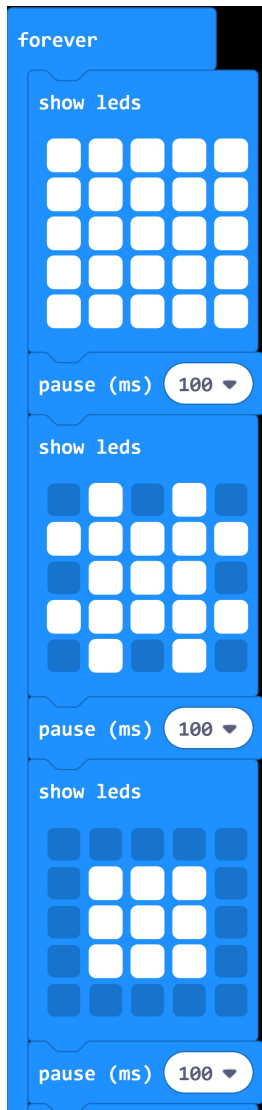
Joonis 6: Microbiti ühendamine toiteploki.

- Salvesta programm (vt kirjeldust eespool)
Fail salvestatakse kohalikku arvutisse laiendiga .hex ja see on programmi kood masinkoodi kujul.
- Ühenda microbit USB juhtmega arvuti külge.
- Kopeeri salvestatud fail microbitti. Kasuta selleks näiteks File Explorerit. Kui tuluke microbiti tagaküljel lõpetab vilkumise, on fail kopeeritud

- Peale kopeerimise lõppu käivitub programm automaatselt. Kui soovid programmi uuesti käivitada, tuleb vajutada microbiti tagaküljel üleval servas USB liidesest paremal olevat nupukest.
- Vajuta nuppe A ja B ning jälgi, mis juhtub.

Kui soovid microbitiga arvutist eemalduda, on vaja eraldi toiteallikat, mis on komplektis olemas väikese patareiploki näol. Vaata joonist (Joonis 6).

LED-paneel



Joonis 7: Erinevate LED piltide vahetamise kood

LED paneelil on 25 LED tulukest ehk leedi (5x5 maatriksina). Leede saab põlema panna ja kustutada. Lihtsam viis microbitiga tutvumiseks on tekitada leedidest erinevaid kujundeid (selleks on eraldi plokk, kus saab kujundi lihtsasti kujundada) ja panna neid vahetuma mõne sündmuse peale (näiteks nuppude vajutamine) või lihtsalt mingi aja tagant. Leedide abil saab ka väljastada teksti: niisama toredaid tervitusi või programmi poolt leitud tulemusi. Seda nägime tegelikult juba nuppude ülesannetes. Võime lasta genereerida juhuslikke arve vahemikus 1..6 ja väita, et microbit töötab nüüd täringuna. Selleks, et programm tööle hakkaks, võib koodi jälle siduda ühe või teise nupu vajutusega.

Ülesanne 3

Kujundame leedidest mõned kujundid ja paneme nad vahetuma mingi aja järel. Ja niimoodi lõputult. Vaata Joonis 7. Plokk **forever** tähendab lõputut tsüklit, mille töö lõppeb siis, kui programmi töö lõpetada. See tsükkel on ennekõike ettenähtud selleks, et ta jookseks teiste tegevuste taustal. Tavalise lõputu tsükli saame plokiga `while true`. Tsüklite plokid leiad sektsioonist **Loop** ning valikuplokid sektsioonist **Logic**. Sektsioonis **Basic** on kaks plokki LED-paneeliga mängimiseks - üks lubab leed-haaval ise pildi kujundada, teise abil saab valida valmis ikoone.

Huvitavama ja programmeerimise mõttes arendavama osa moodustab leedide vilgutamine selliselt, et nad määratakse rea ja veeru koordinaatide abil. See aitab tutvuda ja sõbruneda kahemõõtmeliste massiividega.

Kahemõõtmeliseks massiiviks nimetatakse programmeerimises matemaatikast tuttavaid maatrikseid.

Igal leedil on x ja y koordinaat (x, y). Koordinaat x näitab leedi veeru numbrit ja y rea numbrit. Programmeerimises kutsutakse neid koordinaate **indeksiteks**. Leedide indekseerimine algab ülevalt vasakust nurgast, kus koordinaadiks / indekseks on (0, 0), temast paremal on leed (1, 0), all on leed (0, 1) jne. On olemas plokid leedi sisselülitamiseks (`plot(x, y)`) ja väljalülitamiseks (`unplot(x, y)`). Lisaks

saab ka küsida, kas leed on sisse lülitatud (plokk `point(x, y)`). See plokk (tegelikult küll funktsioon) tagastab väärtuseid `true` või `false` ning annab seeläbi võimaluse kasutada seda valiku- ehk `if`-lauses / plokis. Seda reedab ka ploki kuju.

Ülesanne 4

Proovime leede vilgutada koordinaatide ehk indeksite abil. Vilgutame neid ridu pidi ükshaaval. Vilgutamiseks on vaja leed põlema panna (`plot`), oodata natuke (`pause`) ja seejärel sama leed kustutada (`unplot`). Leedi indeksite määramiseks saame kasutada `for`-tsükli ja tema poolt loodavat muutujat, täpsemalt lausa kahte tsükli, mis lähevad teineteise sisse. Ühe tsükliga määratakse leedi veeru indeks `y`, teisega rea indeks `x`. Vaata koodi Joonis 8.

NB! Muutujad `x` ja `y` tuleb eelnevalt sektsioonis **Variables** valmis teha! `For`-tsükli poolt automaatselt loodav muutuja `index` aitab ühe tsükli korral. Aga meil on vaja kummaski tsükli erinevaid muutujaid leedide eesmärgipäraseks indekseerimiseks.

Ploki on `start` võime vahetada näiteks plokki on `button A pressed` vastu.

Täiendav selgitus for-tsüklile

Tsükkel `for y from 0 to 4` tähendab seda, et tsükli sisu korratakse 5 korda ja selle vältel

muutuja `y` (mida võib nimetada siin kontekstis tsüklimuutujaks) omandab igas korduses uue väärtuse alates 0-st ja lõpetades 4-ga. Seda saame ka väikese eksperimendiga, st programmiga järelle proovida. Kuidas? Lohistad kohale `for`-tsükli plokki ja selle sees trükid välja tsüklimuutujat.

Kui kaks `for`-tsükli on üksteise sees, paneb väline (esimesena algav) tsükkel sisemist tsükli mitu korda tööle. Antud näite puhul viis korda. Tsüklimuutuja `x` sisemises tsükli teeb iga kord läbi muutumise 0-st 4-ni. Meie näites tekivad tsüklimuutujatest `x` ja `y` sisemises tsükli järgmised paarid (`x, y`, esitatud seisemise tsükli korduste kaupa):

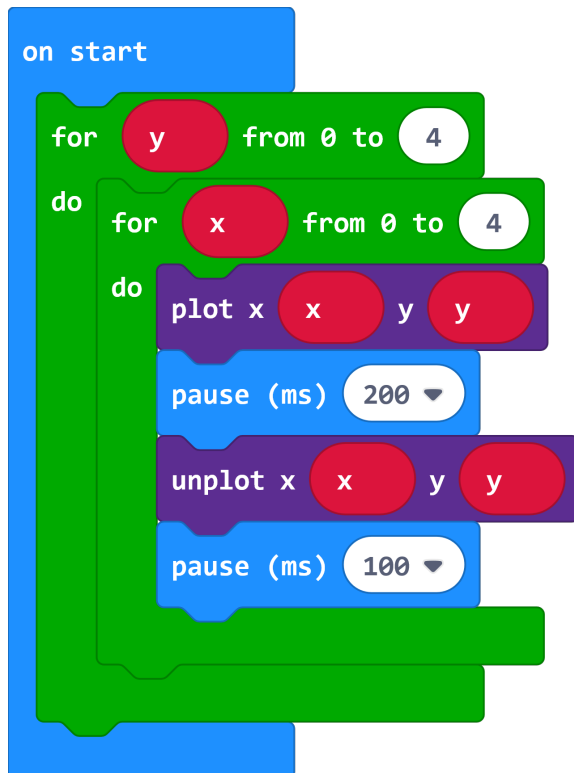
(0, 0), (1, 0), (2, 0), (3, 0), (4, 0)

(0, 1), (1, 1), (2, 1), (3, 1), (4, 1)

(0, 2), (1, 2), (2, 2), (3, 2), (4, 2)

(0, 3), (1, 3), (2, 3), (3, 3), (4, 3)

(0, 4), (1, 4), (2, 4), (3, 4), (4, 4)



Joonis 8: Järjest süttivad ja kustuvad leedid: kahe tsükliga kood

Aga see on täpselt see, kuidas leedid indeksteeritud on. Tänu sellele õnnestub meil ükshaaval iga leedi poole pöörduda.

Veel ülesandeid LED-paneeliga:

1. Mis juhtub, kui x panna välise tsükli muutujaks ning y seesmise tsükli muutujaks ehk vahetada tsüklite alguses x ja y omavahel ära? Proovi, kas Sinu mõte oli õige.
2. Kuidas vilgutada leede mööda peadiagonaali?
3. Pane leedid põlema mööda ridu (ära kustuta!). See võiks toimuda aeglaselt ja rahulikult. Peale seda, kui kõik põlevad, kustuta leedid (aeglaselt ja rahulikult)
 - a) ülevalt vasakult alla paremale mööda veerge;
 - b) alt paremalt üles vasakule mööda ridu;
 - c) ja igasugused muud võimalused. ;)
4. Pane leedid vilkuma suvalises järjekorras ja suvalistes kohtades.
5. Tee leedidest vihmasadu: leedi tuli "liigub ülalt alla". Veerg, milles ta liigub, määratakse juhuslikult.

Servapistmik – 25 väljaviiku

Microbitil on **servapistmik** (*edge connector*) 25 **väljaviiguga** (*pins*), mille külge on võimalik ühendada erinevaid välisseadmeid, et nende abil juhtida programmi tööd. Või vastupidi – juhtida microbitiga teisi seadmeid. Üldistatult on väljaviigud kas sisendi saamiseks microbitile või väljundi andmiseks mõnele microbitiga ühendatud seadmele. Viis viiku on suured, mille külge saab kinnitada erinevaid **klemme** (*clips*), näiteks nn „krokodille”. Numbritega 0, 1 ja 2 tähistatud viigud on sobilikud üldiseks sisendiks ja väljundiks (*GPIO – general purpose input and output*) koos analoog-digitaalmuunduriga (*ADC - analogue to digital converter*). Sinna võib näiteks kinnitada kõrvaklappe.

Viigud 3V ja GND on microbiti varustamiseks voluga. 3V on kolmevoldine sisend või väljund ning GND on maandus. **Neid viikuseid ei tohi mitte kunagi otse kokku ühendada.**

Ülejäänud viikude kasutamiseks on vaja eraldi vidinat nimega *edge connector breakout*, mis annab võimaluse peenemate ühenduste tegemiseks. Viigud on seotud nuppudega, leedidega jne ning kirjeldused leiab microbiti dokumentatsioonist (<https://tech.microbit.org/hardware/edgeconnector/>).

Microbiti külge saab ühendada kõrvaklappe. Selle jaoks on vaja lisaks kõrvaklappidele ka nn "krokodille". Ühendamise juhendi leiad siit lehelt: <https://makecode.microbit.org/projects/hack-your-headphones/make>. Loomulikult on ilusam kasutada ühendamiseks spetsiaalset vidinat, mis kahest krokodillitsast ühe kõrvaklapiotsa tekitab.

Muusikaplokid

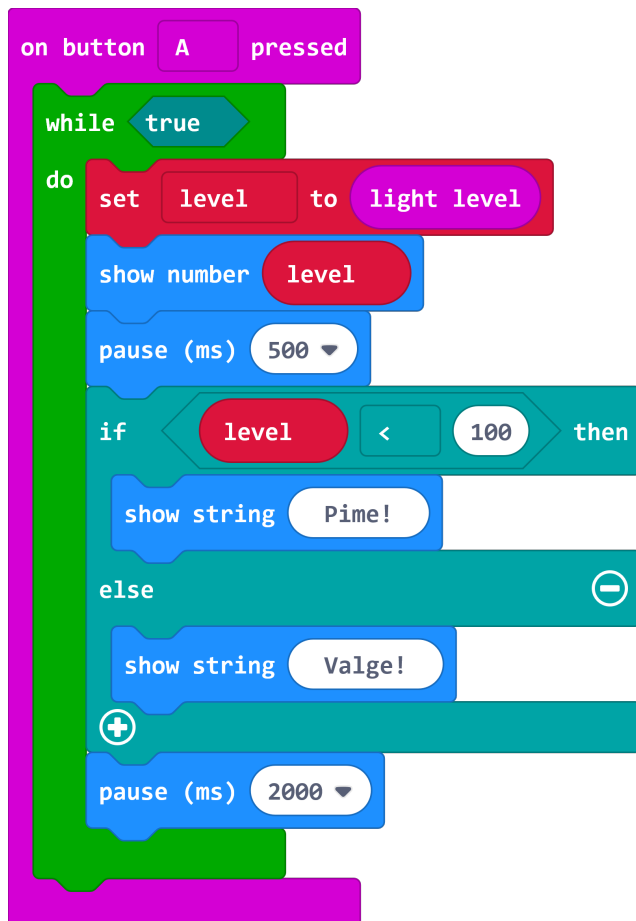
Koos kõrvaklappide kasutamisega avaneb võimalus ka natuke muusikat teha. Selleks on plokkide sektsioon **Music**. Lihtsamal juhul saab järjest lasta noote mängida ja anda neile erinevaid pikkuseid (`play tone`). Ka paus on olemas (`rest (ms)`). Sama fraasi kordamiseks saab kasutada tsüklit ("*Rongisõit*", "*Juba linnukesed ...*"). Samuti saab lasta mängida mõnda eelkirjeldatud meloodiat (`start melody`).

Microbiti andurid

Microbiti plaadil on mitmed andurid, millega saab jälgida ümbritsevat olukorda ... edukamalt või vähem edukalt. Samuti saab jälgida microbiti asendit ja tema liikumist. Jälgida saab ümbritsevat valgust, temperatuuri, microbiti asendi muutust x, y ja z telgede suhtes ning suunda maa magnetvälja suhtes.

Anduritelt väärtuste lugemiseks ei ole eraldi käske, vaid selleks on erinevad sisendiplokid, mille leiab sektsioonist **Input**. Loetud sisendi saab omistada muutujasse ning edasi muutujat kasutada soovitud eesmärgil.

Ümbritsev valgus



Joonis 9: Valguse mõõtmise ja hinnang

Microbitiga on võimalik hinnata ümbritsevat valgust. Siiski päris valgusandurit microbiti küljes ei ole. Anduritena kasutatakse leede. Ümbritsevat valgust hinnatakse skaalal 0 (täitsa pime) kuni 255 (päris valge). Mida sellega teha saab? Näiteks näidata nupu vajutamise peale valgustatust. Panna paika mingid piirid ja kommenteerida kui valge on. Aga miks mitte ka ühendada microbit mõne välisseadmega ja valguse muutumisel lülitada sisse lamp?

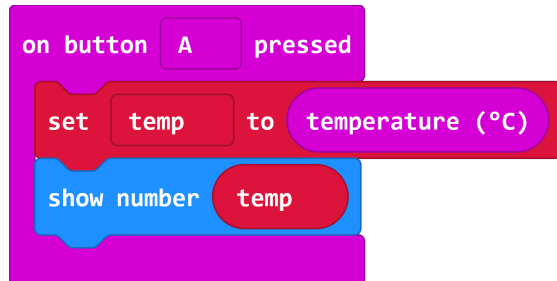
Valguse arvulise väärtuse saab sisendiplokist `light level`. Selle saab omistada muutujasse ning edasi kasutada muutuja väärtust, nagu selleks soov on.

Ülesanne 5

Nupu A vajutuse peale käivitub lõputu tsükel, kus iga paari sekundi järel fikseeritakse valgus, ekraanile kuvatakse vastav number ning kommentaar, kas on pime või valge. Piir (mis on siin kohal täiesti suvaliselt valitud) on näites 100. Vaata ka koodi (Joonis 9). Nüüd võib microbiti

pihku võtta ja sellega ringi jalutada ning vaadata, kuidas valgus muutub. Kuna mõõtmine toimub leedide abil, ei tohiks neid käega kinni katta. Või siis vastupidi - leede varjutades, saab vaadata, mis juhtub valgusega.

Termoandur ehk temperatuuriandur



Joonis 10: Temperatuuri mõõtmine

Microbitil on sisseehitatud termoandur. Andur mõõdab seadme, täpsemalt protsessori temperatuuri. Väidetavalt annab see enam-vähem õigesti edasi ümbritseva temperatuuri. Seda siiski eeldusel, et protsessor ei ole eelnevalt väga kõvasti tööd teinud, mis oleks tema temperatuuri tõstnud. Samuti ei reageeri seade kiirele temperatuuri muutusele. Enne peab ta jõudma ise uue ümbritseva temperatuuri omandada. Kood on sarnane valguse mõõtmisele (vt Joonis 10). Seekord saadakse

väärtus sisendiplokist `temperature (°C)`.

Kiirendusandur ehk kiirendusmõõtur ehk aktseleromeeter

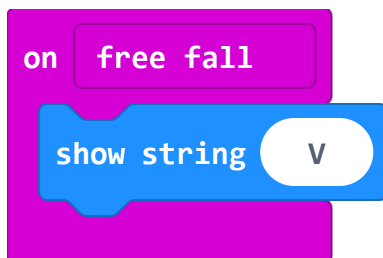
Kiirendusanduri leiad microbiti tagaküljelt alt vasakust nurgast (vt oma microbitti ja Joonis 1). Seega siin on tegu juba spetsiaalse mikroskeemi ehk kiibiga. Kiirendusandur fikseerib microbiti liikumist.

Sarnane kiip on ka näiteks mobiiltelefonide sees, tänu millele aparaat "saab aru", kui teda keeratakse ja ta siis pildi teist pidi pöörab.

Microbit on programmeeritud mitmeid erinevaid liigutusi ära tundma ning nende jaoks on kirjeldatud sündmused. Vastavaid sündmusi saab kinni püüda plokiga **ON** seksioonist *Input*. Vast kõige rohkem ekspluateeritud sündmusteks on **raputamine** (*shake*). Aga kirjeldatud on ka sündmused **vabalangemine** (*freefall*), **kallutus vasakule** või **paremale** (*tilt left, right*), **pööramine** (*logo up, logo down*) ning **raskuskiirendust** (3g, 8g). Kuidas need sündmused täpsemalt avalduvad, vajab microbitiga proovimist.

Sündmust omakorda saab kasutada mingi tegevuse (koodiploki) käivitamiseks. Siin kohal ongi raputamine üleekspluateeritud. Mõned näited:

Kui raputatakse seadet, genereeri juhuslik arv ja näita seda ekraanil.

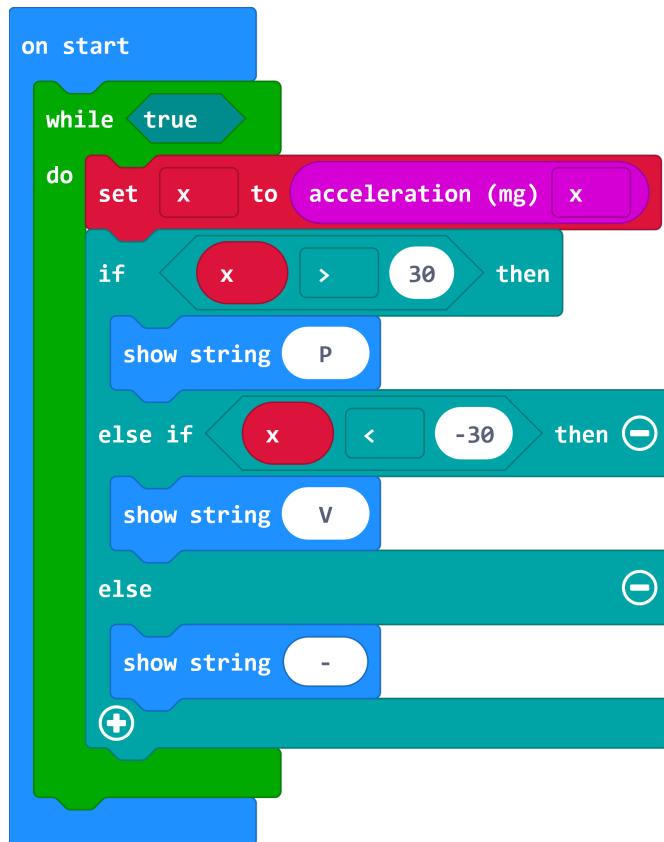


Joonis 11: Viipe katsetus, lisa teisi sündmuseid

Mäng „kivi, paber, käärid”. Kui raputatakse seadet, genereeri juhuslik arv 0..2. Igal numbril on tähendus (kas kivi, paber või käärid). Kuva sobiv kujund leedpaneelile. Nüüd leia microbittidega sõbrad ja mängi.

Uurimaks täpsemalt, mida üks või teine liigutusega seotud sündmus tähendab, võid teha katseprogrammi, kus paned microbiti erinevate sündmuste puhul erinevaid sümboleid näitama (vt Joonis 11). Selleks tuleb sarnaseid plokkide oma koodiaknasse mitu tükki teha. Aga saab

moodustada ka ühe koodi, kus tsükli sees `if`-plokki kasudes lastakse vastavalt viipe iseloomule (näiteks `is shake gesture`) näidata LED-paneelil erinevaid tähti. Raskuskiirenduse saavutamiseks jaoks ei ole soovitatav teda vastu põrandat või seina visata, saab ka käes kiiret pidurdamist katsetada. Samuti vabalangemist. Kiirema käes pidurduse korral sain kätte ka sündmuse `8g`.



Joonis 12: Koordinaadid kallutamisel

kohal teine väike kiip on kompass. Kompass kasutab maa magnetvälja ja selle järgi on võimalik öelda, mis pidi microbit parasjagu on. Kompass tuleb enne kasutusele võtmist kalibreerida, et tagada tulemuste korrektsus. Seda palub seade ise, kui kompassi lugema hakatakse. Kalibreerimisest saab lugeda ja vaadata siin: <https://support.microbit.org/support/solutions/articles/19000008874-calibrating-the-micro-bit-compass>

Ülesanne 6

Kui on soov täpsemalt teada saada, mida kiirendusmõõtur välja loeb, saab katsetada eraldi `x`, `y` ja `z` koordinaatide väärtuste välja lugemisega. Võimaluse selleks leiad `input`-plokkide alt. Sisend saadakse spetsiaalsest plokist `acceleration (mg) x`.

Loeme kiirendusmõõturi poolt tagastatavat `x`-i väärtust ja anname teada, kas microbit on `x`-telje suhtes kallutatud paremale (`P`), vasakule (`V`) või ei ole kallutatud (`-`). Koodi vaata siit (Joonis 12).

Võimalik, et neid koordinaate kasutades saab õpetada ja katsetada midagi vektorite kohta. Seda oskavad täpsemalt pakkuda vektoritega paremini kursis olevad inimesed.

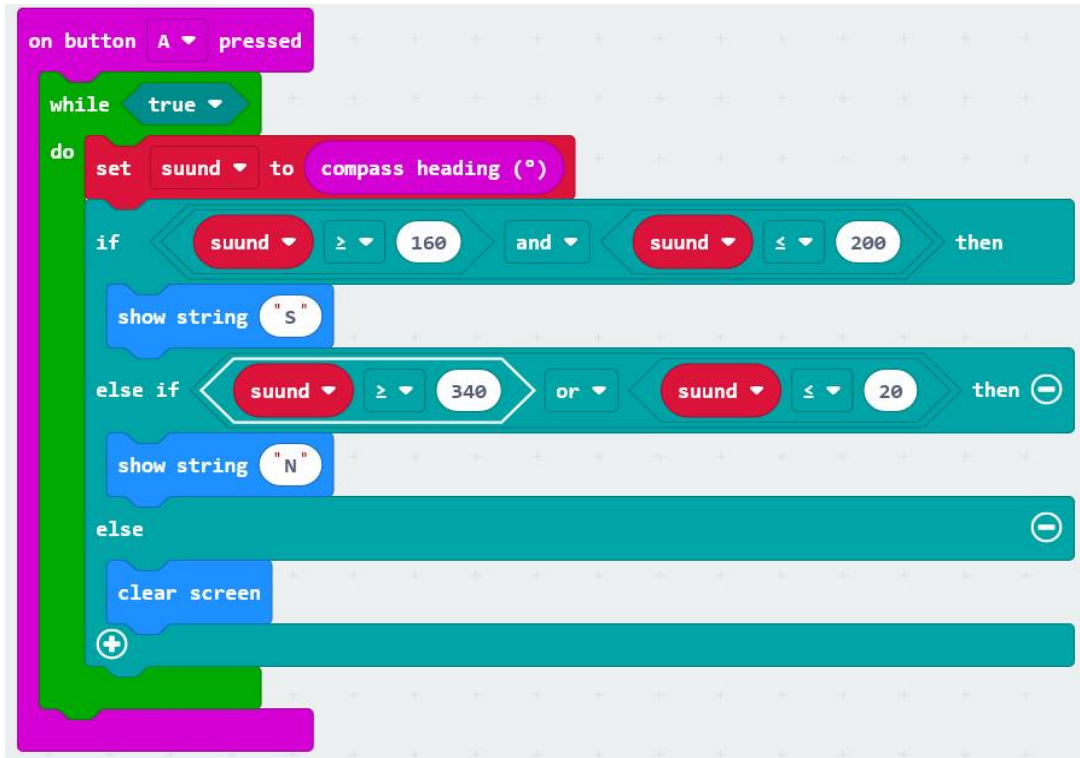
Kompass

Microbiti all vasakus nurgas kiirendusmõõturi kohal teine väike kiip on kompass. Kompass kasutab maa magnetvälja ja selle järgi on võimalik öelda,



Joonis 13: Kompassi ja micribiti suuna võrdlus.

Plokk compass heading (°) annab täisarvulise väärtuse vahemikus 0 .. 359, mis peaks näitama kompassi (microbiti) suunda kraadides maa magnetvälja suhtes. Põhjasuund on 0°. Praktilised katsetused andsid sellise pildi (Joonis 13), millelt on näha, et microbiti poolt näidatav kraad ei ühti päriselt kompassi magnetnõelaga.



Joonis 14: Kompass - põhja-lõuna suuna määramine

Ülesanne 7

Tüüpiline ülesanne on muuta microbit kompassiks, mis siis kas noolelaadsete kujunditega või tähti kasutades ütleb, kuhu suunda seade suunatud on. Suunad tuleb siduda kraadidega ja määrata vahemikud, mida loeme vastavalt põhja-, lõuna- jne suunaks. Teeme järgmised otsused: microbit näitab nelja põhilist ilmakaart, kasutades tähti N, O, S ja W. Suunda näitab ta siis, kui erinevus täpsest suunast on +/-20 kraadi. Kui põhi (N) on 0°, siis lõuna (S) on 180°. Arvuta välja ka ida ja lääs. Näitekoodis (vt Joonis 14) on valik vaid põhja ja lõuna vahel.

Ülesanne 8

Proovi teha mäng, mis õpetab ilmakaari tundma. Selleks on vajalikud järgmised sammud:

Loosi üks ilmakaar (N, E/O, S, W) ja kuva vastav täht ekraanil. Vihje – tee juhuslik arv 0 .. 3 ja seo iga arv ühe ilmakaarega (`if`).

Jälgi microbiti suunda, kuni ta keeratakse nõ õigeks. Vihje – `while`-tsükkel, mis töötab seni, kuni suund ei ole vajalikus vahemikus.

Anna õigeks pööratud microbitist mingil viisil leedekraanil märku.

Võid lisada ka vaheilmakaared. Võid proovida mõõta aega, mis kulub microbiti õigeks seadmisele. Selleks on `input` plokkide hulgas funktsioonid `running time (ms)` või `running time (micros)`. Nad annavad tulemuseks aja microbiti sisselülitumisest millisekundites või mikrosekundites. Mõõtes kaks korda aega, kõigepealt peale ilmakaare loosimist ja seejärel peale mängijapoolset õige suuna kättesaamist, saame aegade vahest teada kulunud aja. Siin on vaja kasutada muutujaid.

Raadioühendus ja Bluetooth

Microbit saab infot vahetada teiste seadmetega. Seadmel on 2.4 GHz raadiosaatja-vastuvõtja (*transceiver*), mida saab kasutada nii Bluetooth'i abil suhtlemiseks kui ka madalamal tasemel raadioühenduseks.

Raadioühenduse abil saab juhtmevabalt suhelda näiteks teiste microbittidega. Bluetoothi kaudu saab microbit suhelda Bluetooth-seadmetega, sh nutitelefonide ja tahvelarvutitega. Bluetooth ühenduse korral luuakse sidekanal kahe seadme vahel (nad nõ pairitatakse), mis on turvaline ja suhtlemine toimub just nende kahe seadme vahel. Raadioühendus on lihtsam. Seade saadab lihtsalt sõnumi ja loodab, et keegi selle kätte saab.

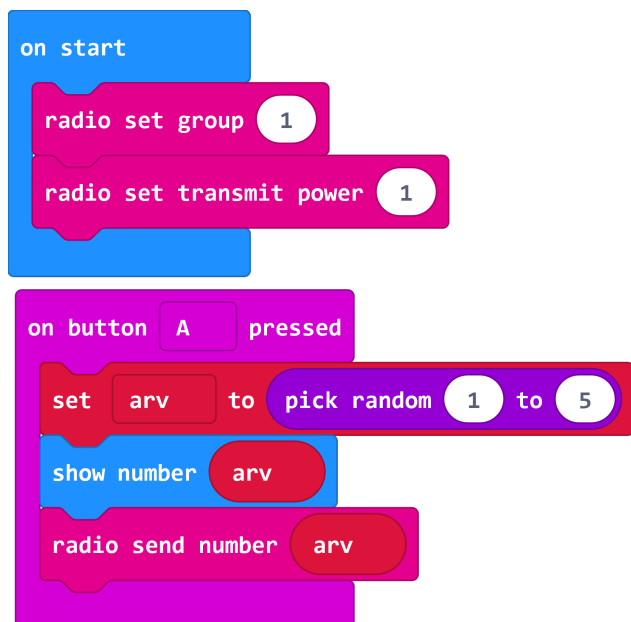
Raadioühenduse loomisel saab microbitid ühendada gruppidesse plokiga `radio set group nr.` Kui on seadistatud grupi number, siis filtreeritakse microbiti poolt automaatselt välja õiged sõnumid ehk teise grupi microbittide sõnumeid ignoreeritakse. Kui grupp ei ole seadistatud, sõnumite vastuvõtmisel piiranguid ei ole. Saata on võimalik arvu (`radio send number`), teksti (`radio send string`) ja nime ning väärtuse paari (`radio send value`). Kahe viimasega on võimalik

kaasa panna ka seadme seerianumber, kui selle saatmine on sisselülitatud plokiga `radio set transmit serial number`. Selle abil on võimalik eristada sõnumeid saatnud microbitte.

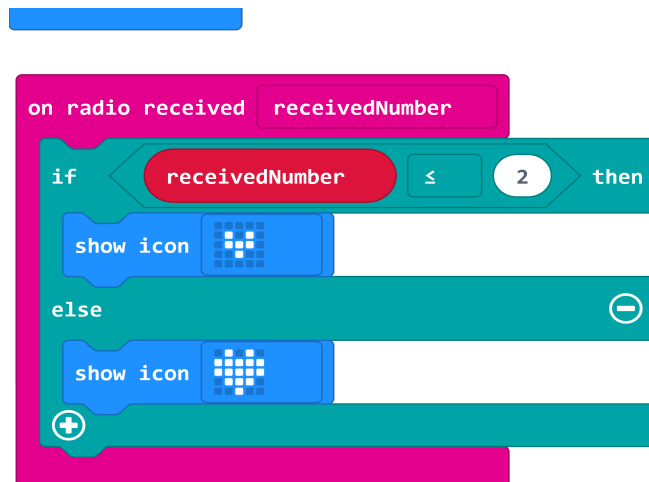
Microbitti on võimalik panna tegema midagi, kui ta püüab kinni teise saadetud info. Selleks on plokk `radio received`. Saadetud sõnumi sisu on plokis `received packet`.

Ülesanne 9

Paneme kaks microbitti suhtlema nii, et nad saaksid teineteisele oma tundeid väljendada. Üks microbit loosib arvu (näites 1..5) ja saadab selle laiali. Teine microbit püüab arvu kinni ja kui see arv on 1..2, siis ta saab aru, et teda armastatakse natuke ning ta näitab väikest südant. Kui aga arv on suurem, siis ta näitab suurt südant. Plokis `on start` on grupi määramine ja raadiosaatja/vastuvõtja võimsuse määramine (kui hädavajalik viimane plokk on, pole ma aru saanud). Microbit saadab oma tunded teele, kui vajutad nuppu A (vt Joonis 16) ning plokki `radio received` abil reageerib kinnipüütud arvule (vaata Joonis 15). Kogu koodi võib panna mõlemasse microbitti, siis nad saavad vastastikku teateid saata ja reageerida.



Joonis 16: Microbiti seadistamine ja arvu saatmine.



Joonis 15: Reageerimine sõnumile.

Ülesanne 10

Veel mõned internetist leitud ideed, millega katsetada

Ettevaatust, varas tuleb!!

Pane microbit ukse külge. Pane te reageerima liikumisele (proovi järgi, milline töötab). Kui ust liigutatakse, saada raadio teel mingi teade teisele microbitile ja lase sellel kuidagi reageerida – leedide vilgutamisega, helisignaaliga vms.

Jalgratta suunatuli ja stopptuli.

Kui vajutad ühel microbitil nuppu B, süttib teisel nool paremale, kui vajutad nuppu A, süttib nool vasakule. Kui vajutad mõlemat tuleb mingi stoppmärk.

Kodune ülesanne

Siin materjalis olid mõned ülesanded, millel ei olnud koodi juures (näiteks kompassimäng ajapeale, raadioühenduse kaks ideed, ...). Vali nendest midagi ja tee kodutööks. Võid otsida ka teisi ideid. Aga tore oleks, kui Sa ise midagi teed või vähemalt mingil viisil muudad, mitte ei esita internetist leitud projekti.

Esitamiseks salvesta fail, anna talle nimeks oma nimi (maali_maasikas.hex) ja saada mulle meilile. Selgita ka meilis, mida programm teeb. Kui kasutasid ideid internetist, siis lisa ka nende lingid (ehk viited algallikatele).