



Primjeri za rad s Boson setom za micro:bit



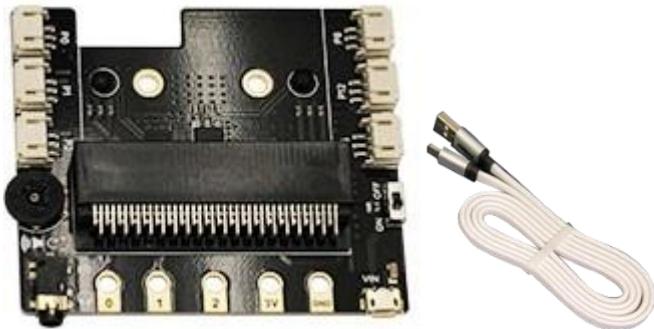
Sadržaj:

Dijelovi Boson seta	2
Projekt 1: Svjetlo obavijesti.....	4
Projekt 2: Reguliranje intenziteta svjetla.....	11
Projekt 3: Ventilator	20
Projekt 4: Mjerač zvuka	34
Projekt 5: Poruka za sreću	42
Projekt 6: Uređaj za praćenje učenja i pauze.....	50
Projekt 7: Disco light.....	63
Projekt 8: Rampa na parkiralištu.....	70
Projekt 9: Roboalarm.....	75
Projekt 10: Robot – kreiraj ga micro:bit kockicom	86

Dijelovi Boson seta

Pločica za spajanje elektroničkih modula i USB kabel za napajanje

Omogućuje lakše povezivanje micro:bita s modulima.



Kablovi za spajanje modula s pločicom

Služe za fizičko povezivanje modula na pločicu.



Tipkalo

Digitalni ulazni modul

Pokreće ili zaustavlja rad drugih modula.



Senzor pokreta

Digitalni ulazni modul

Prepoznaje kretanje.



Senzor zvuka

Analogni ulazni modul

Detektira intenzitet zvuka u prostoru.



Senzor rotacije (potenciometar)

Analogni ulazni modul

Služi za promjenu vrijednosti od minimalne do maksimalne (npr. reguliranje glasnoće na radiju).



Svjetleća dioda

Digitalni/analogni izlazni modul

Svijetli kad dobije signal od ulaznih modula.



Ventilator

Digitalni/analogni izlazni modul

Vrti se kad dobije signal od ulaznih modula.



Servo motor

Može zakretati polugice za točno određeni kut.



RGB LED traka

Svijetli određenim brojem dioda i različitim bojama kad dobije signal od ulaznih modula.



Projekt 1: Svjetlo obavijesti

Opis projekta

Za početak upoznavanja s Boson setom izradite projekt *Svjetlo obavijesti* u kojem ćete upotrijebiti tipkalo i svjetleću diodu. U ovom projektu ćete naučiti kako spojiti module na pločicu kablovima za spajanje te uočiti odnos između rada ulaznog i izlaznog modula. U projektu *Svjetlo obavijesti* tipkalo je digitalni ulazni modul,a svjetleća dioda je digitalni izlazni modul. Svaki put kad se pritisne tipkalo, svjetleća dioda će zasvijetliti kao znak (obavijest) da je tipkalo pritisnuto. Dodatno, izradite ovaj projekt tako da se u isto vrijeme kao i vanjska svjetleća dioda, i na micro:bita pali i gasi dioda koja se nalazi na sredini zaslona (pozicija: 2, 2).

Pomoću ovih modula možete izraditi različite programe i kreativne projekte. Primjerice, tipkalo možete pretvoriti u sklopku koja će paliti i gasiti svjetleću diodu, pritiskom na tipkalo možete regulirati intenzitet kojog dioda svijetli te možete napraviti program kojim će se svjetleća dioda paliti i gasiti u određenom ritmu kada pritisnete tipkalo.

Pogledajte u kreativnom kutku ovog projekta ideju u kojoj su tipkalo i svjetleća dioda dio dostave cvijeća u cvjećarni izrađenog od lego kockica.

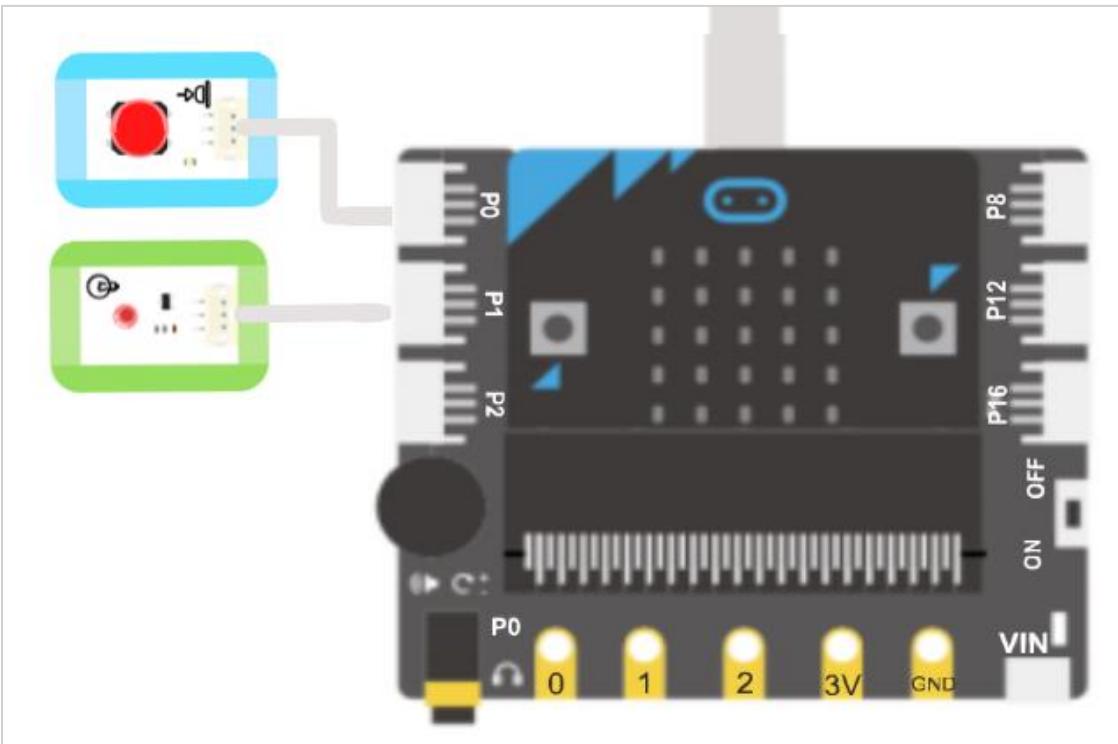
Potrebni moduli

Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 2 kabela za spajanje modula, tipkalo, svjetleća dioda

Spajanje modula

U pločicu smjestite micro:bit. Na mjestima za spajanje elektroničkih modula kablovima povežite i spojite tipkalo te svjetleću diodu kao što vidite na prikazu spoja. Tipkalo je digitalni ulazni modul pa se može spojiti na bilo koji izvod na pločici. Svjetleća dioda može biti digitalni ili analogni izlazni modul te se isto može spojiti na bilo koji izvod. U ovom primjeru, prema montažnoj shemi, tipkalo je spojeno na izvod P0, a svjetleća dioda na izvod P1. USB kabelom povežite micro:bit s računalom kako bi spoj dobio napajanje i kako biste prebacili program na micro:bit.

Prikaz spajanja



Izrada programa

Potrebne naredbe: *forever, if then else, operator za usporedbu veličina, digital read pin P0, digital write pin P0 to 0, show leds, clear screen*

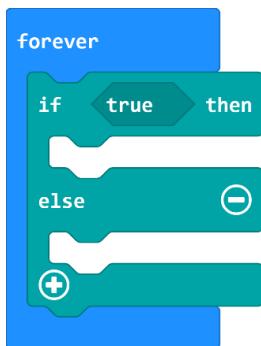
Sve naredbe u ovom programu smjestite u *forever* petlju jer želite da se program neprestano izvršava. To jest, da svaki put kad pritisnete na tipkalo, upalite svjetleću diodu. *Forever* petlja nalazi se u kategoriji

Basic.

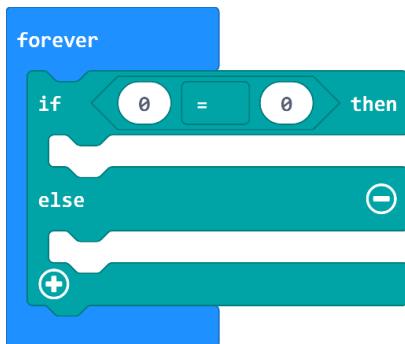


Slijedi dio u kojem ćete definirati uvjet koji mora biti ostvaren kako bi svjetleća dioda zasvijetlila. Uvjet za to je pritisak na tipkalo. Tipkalo je digitalni ulazni modul spojen na izvod P0. Svaki put kad se pritisne tipkalo, izvod P0 poprima vrijednost 1. Kad se tipkalo otpusti, vrijednost izvoda P0 je 0. Vrijednost izvoda P0 na kojem je spojeno tipkalo (digitalni ulazni modul) micro:bit očitava.

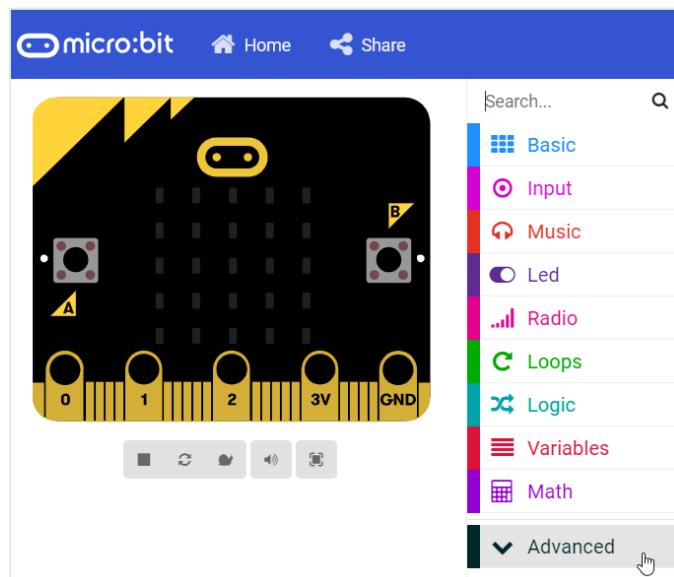
Iz kategorije **Logic** odaberite naredbu *if then else* i smjestite ju u *forever* petlju.



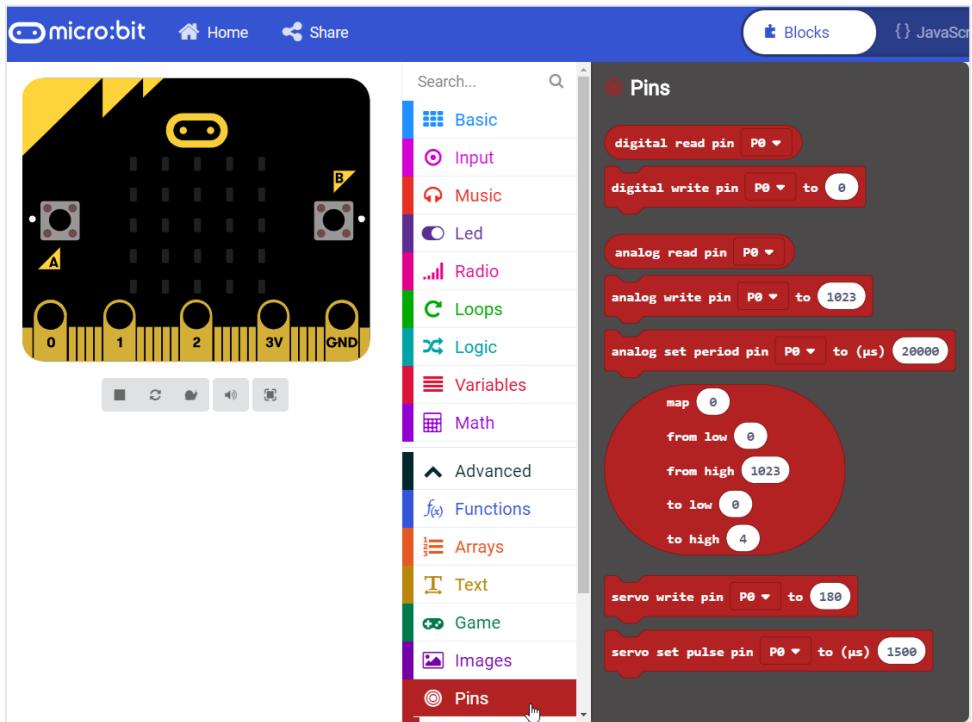
U *if* dijelu naredbe postavite operator za usporedbu (jednako) koja se nalazi u kategoriji **Logic**.



Time ćete moći provjeriti uvjet koji mora glasiti: očitana vrijednost na izvodu P0 je jednaka 1. S lijeve strane usporedbe potrebno je postaviti blok koji predstavlja digitalno očitanu vrijednost izvoda P0. Taj blok se nalazi u kategoriji **Pins**. **Pins** kategoriju ćete pronaći u **Advanced** dijelu kategorija kad ga otvorite.



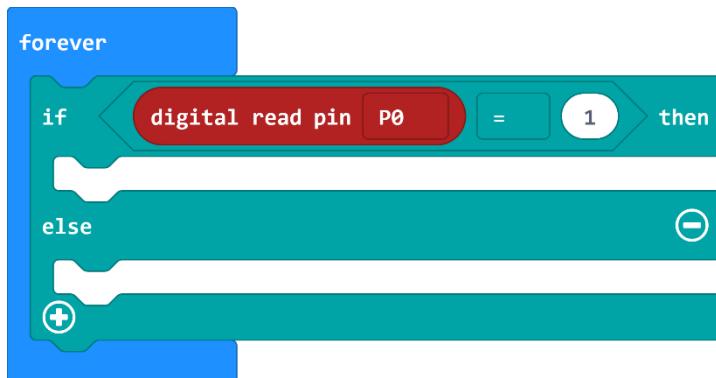
Pritiskom na kategoriju **Pins** otvara se popis naredbi koje ona sadrži.



Odaberite prvi blok *digital read pin P0*. Njime micro:bit provjerava digitalni signal na izvodu P0.

digital read pin P0

Taj blok postavite s lijeve strane usporedbe, a s desne upišite broj 1. Sada je definiran uvjet kojim se provjerava je li vrijednost digitalnog signala na izvodu P0 jednaka 1. Drugim riječima, je li pritisnuto tipkalo u spoju.

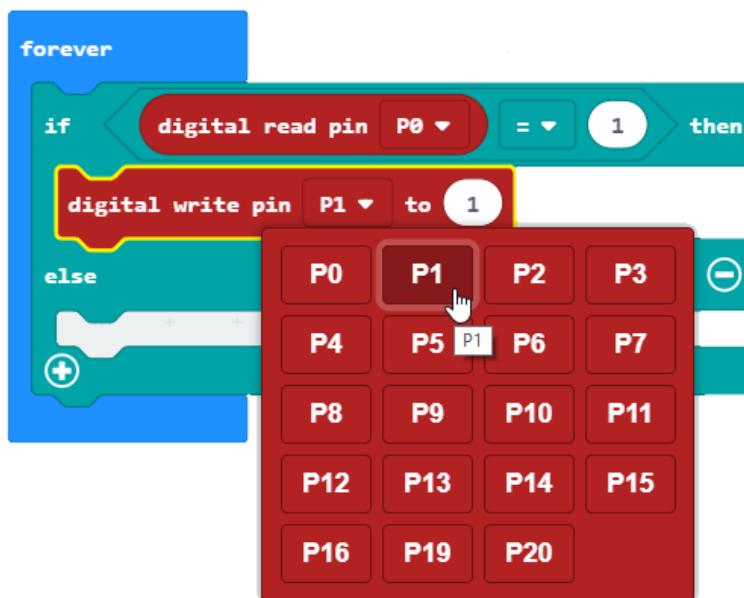


Ako je uvjet ostvaren (tipkalo je pritisnuto), svjetleća dioda mora zasvijetliti. To ćete u programskome kôdu definirati u then dijelu naredbe. Svjetleća dioda je u ovom spoju digitalni izlazni modul spojen na izvod P1. Preko izvoda P1 svjetleća dioda će primiti digitalni signal. Micro:bit kontrolira taj proces prateći stanje na izvodu P0. Kada micro:bit s izvoda P0 dobije digitalni signal 1, tada na izvodu P1 postavlja vrijednost digitalnog signala na 1.

Naredba kojim micro:bit na izvode postavlja vrijednost digitalnog signala nalazi se u kategoriji **Pins** i zove se *digital write pin P0 to 0*. U toj naredbi možete odabrati izvod koji vam treba te upisati vrijednost digitalnog signala koja može biti 0 ili 1.



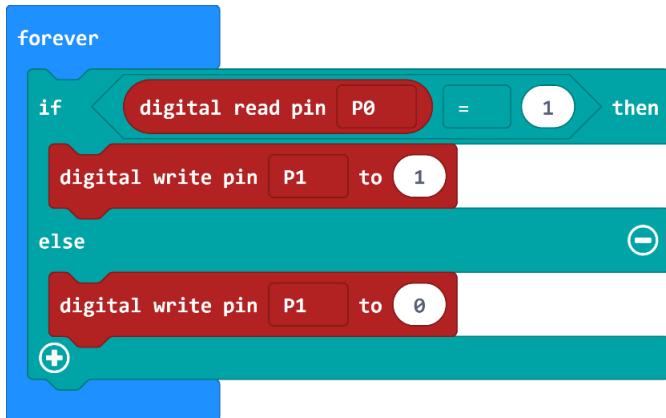
Odaberite naredbu *digital write pin P0 to 0* i smjestite ju u *then* dio naredbe. Pritiskom na strelicu uz parametar P0, izmjenite P0 u P1 te umjesto 0 kao argument naredbe upišite 1.



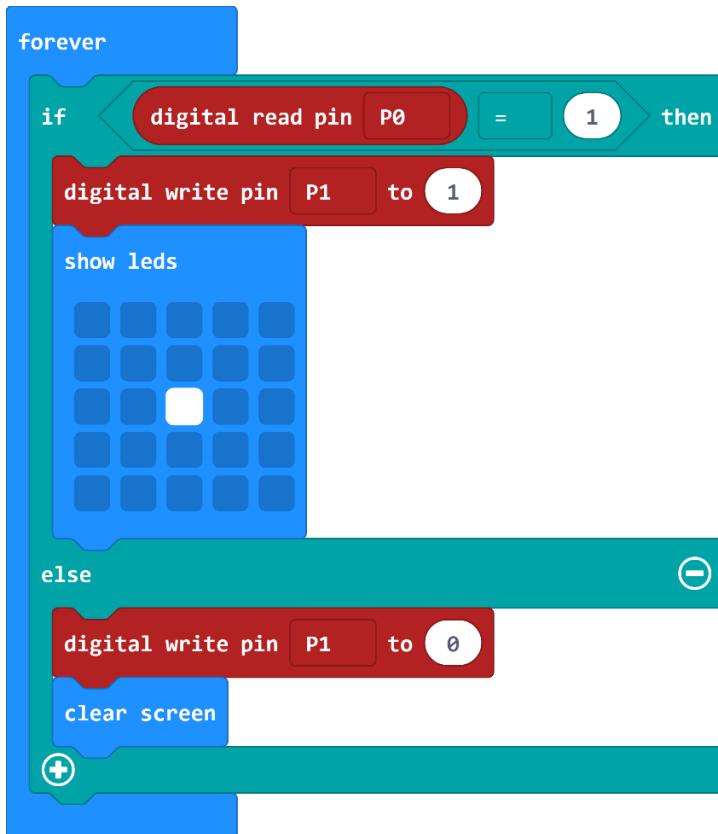
Time je definirano ako je vrijednost digitalnog signala na izvodu P0 jednaka 1 (tipkalo je pritisnuto), tada izvod P1 poprima vrijednost 1 (svjetleća dioda svijetli).

U *else* dijelu sada valja definirati što će se događati kada uvjet nije ostvaren (kad tipkalo nije pritisnuto). Kad tipkalo nije pritisnuto, svjetleća dioda neće svijetliti. To znači da će stanje izvoda P1 na kojem je

dioda spojena biti 0. U else dio naredbe postavite naredbu digital write pin P0 to 0 te umjesto P0 odaberite P1.



I na samom micro:bitu možete pratiti rad tipkala i svjetleće diode. U *if* dio dodajte naredbu *show leds* iz kategorije **Basic** i u njoj odaberite diodu koja se nalazi na sredini zaslona. U *else* dio dodajte naredbu *clear screen* iz kategorije **Basic** u dijelu more. Tim naredbama će se i na zaslonu micro:bita paliti i gasiti dioda ovisno o tome je li tipkalo pritisnuto. Primijetite da svjetleće diode (i vanjska i ona na micro:bitu) svijetle onoliko dugo koliko je tipkalo pritisnuto.



Kreativni kutak

Bosom elementima iz ovog projekta te lego kockicama možete izraditi dostavu cvijeća u cvjećarnu. Kad dostavljačica dođe do cvjećarne tipkalom će javiti da je tu, a cvjećaru će se upaliti crvena lampica kao obavijest da je dostava stigla.

Osim lego kockicama, kreativni dio ovog projekta možete izraditi pomoću kartona, kolaža, flomastera, škara i ljepila te drugih materijala.



Projekt 2: Reguliranje intenziteta svjetla

Opis projekta

U ovom projektu upoznat ćete analogni ulazni modul potenciometar. Pomoću potenciometra ćete regulirati intenzitet svjetla svjetleće diode koja će u ovom projektu biti analogni izlazni modul. Ovisno o položaju potenciometra, svjetleća dioda će svijetliti jače ili slabije. Neka se na zaslonu micro:bita ispisuje vrijednost analognog signala koji micro:bit prima od potenciometra.

Dodatno, možete regulirati i rad svjetlećih dioda na zaslonu micro:bita i poigrati se s mogućnostima. Najprije izradite program kojim će svjetleće diode na micro:bitu svijetliti jednakim intenzitetom kao vanjska svjetleća dioda, a zatim napravite program kojim će svjetleće diode na micro:bitu svijetliti jače kada vanjska dioda svijetli slabije i obrnuto.

U kreativnom kutku pogledajte kako ove module i program iskoristiti za izradu origami lampe.

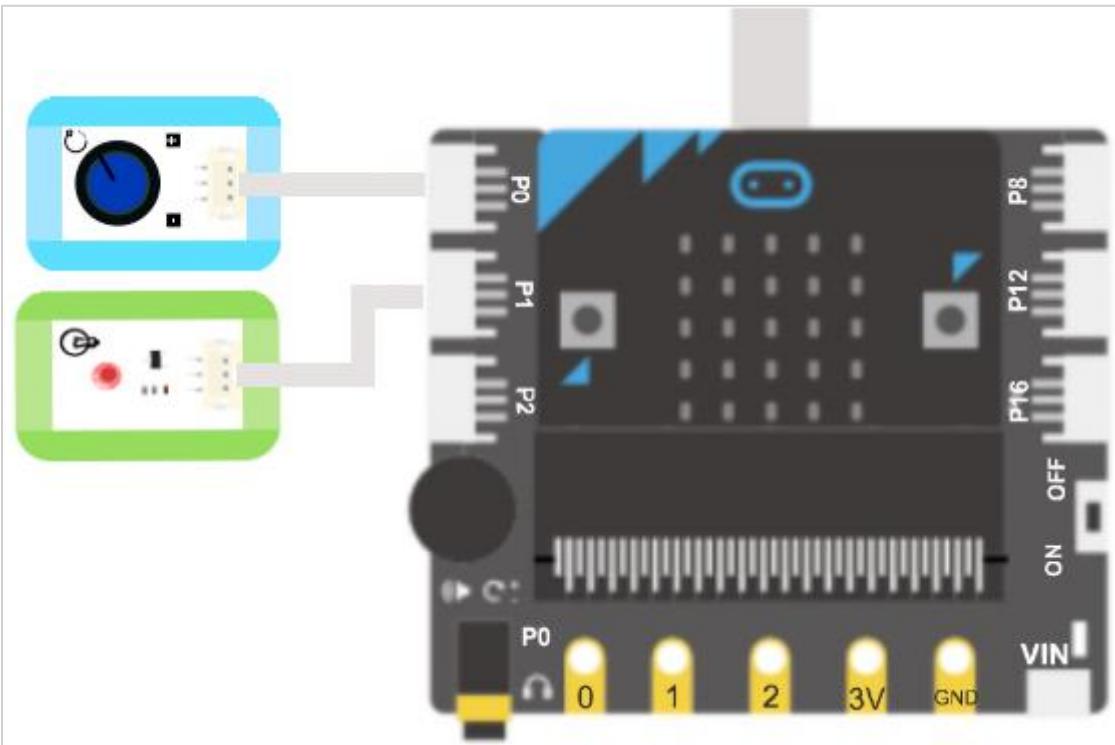
Potrebni moduli

Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 2 kabela za spajanje modula, potenciometar, svjetleća dioda

Spajanje modula

U pločicu smjestite micro:bit. Na mjestima za spajanje elektroničkih modula kablovima povežite i spojite potenciometar te svjetleću diodu kao što vidite na prikazu spoja. Potenciometar je analogni ulazni modul pa se može spojiti samo na izvode koji su analogni. To su izvodi P0, P1 ili P2 na pločici. Micro:bit preko izvoda na kojem je spojen potenciometar kao analogni ulazni modul, očitava stanje njegovog signala čiji se raspon kreće u vrijednostima od 0 do 1023. Već ste naučili da svjetleća dioda može biti digitalni ili analogni izlazni modul. U ovom projektu će svjetleća dioda biti analogni izlazni modul što znači da ju se može spojiti samo na izvode P0, P1 ili P2. Ovisno o vrijednosti analognog signala koji će svjetleća dioda primiti preko svog izvoda, svijetlit će jačim ili slabijim intenzitetom. Što je veća vrijednost signala, to dioda svijetli jače. Prema prikazu spajanja, potenciometar je spojeno na izvod P0, a svjetleća dioda na izvod P1. USB kabelom povežite micro:bit s računalom kako bi spoj dobio napajanje i kako biste prebacili program na micro:bit.

Prikaz spajanja



Izrada programa

Potrebne naredbe: *forever, analog write pin P0 to 1023, analog read pin P0, show number, set brightness, show leds, map*

Sve naredbe u ovom programu smjestite u *forever* petlju jer želite da se program neprestano izvršava. Tako ćete cijelo vrijeme moći pratiti promjene u intenzitetu rada svjetleće diode dok okrećete. *Forever* petlja nalazi se u kategoriji **Basic**.



S obzirom na to da je svjetleća dioda ovaj put analogni izlazni modul, micro:bit na izvodu P1 na kojem je ona spojena postavlja analognu vrijednost od 0 do 1023 koju očitava od analognog ulaznog modula, odnosno potenciometra.

U kategoriji **Pins** odaberite naredbu *analog write pin P0 to 1023*. Ovom naredbom definirat ćete vrijednost signala koji će izvod P1 primiti od micro:bita i prikazati u obliku intenziteta rada diode.



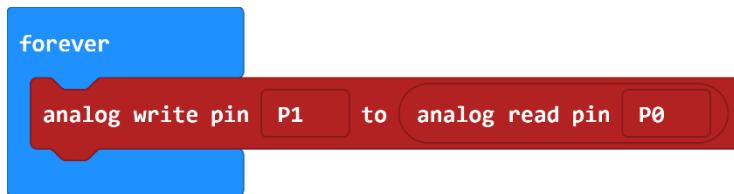
Naredbu smjestite u *forever* petlju. Pritiskom na strelicu uz parametar P0, odaberite P1 jer je na njemu spojena svjetleća dioda.



Trenutno u programu stoji da se na analogni izvod P1 postavlja vrijednost signala na maksimalnu, 1023. Vama za ovaj projekt treba da se na izvod P1 postavlja vrijednost signala koju regulira potenciometar. Kao i digitalne signale, micro:bit može očitavati i analogne signale. Iz kategorije **Pins** odaberite naredbu *analog read pin P0*.



Naredba *analog read pin P0* predstavlja vrijednost signala na izvodu P0. Na tom izvodu je spojen potenciometar. Ovisno o njegovom položaju micro:bit očitava različitu vrijednost signala. Ako je potenciometar okrenut prema minusu, signal je manji. Ako je okrenut prema plusu, signal je veći. Smjestite naredbu *analog read pin P0* kao argument naredbe *analog write pin P1* umjesto broja 1023.

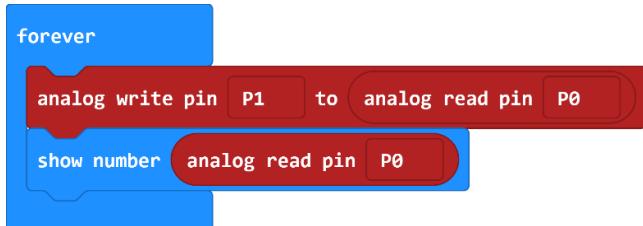


Ovim programom stvorena je veza između vrijednosti signala analognog ulaza (potenciometra) i vrijednosti signala analognog izlaza (svjetleće diode). Upravo ona vrijednost koja se očita na ulaznom modulu, postavlja se na izlazni. Ta vrijednost može se provjeriti ako u program dodate naredbu za njezin prikaz na zaslonu micro:bita.

Iz kategorije **Basic** odaberite prvu naredbu, *show number*. Naredbu smjestite u *forever* petlju.



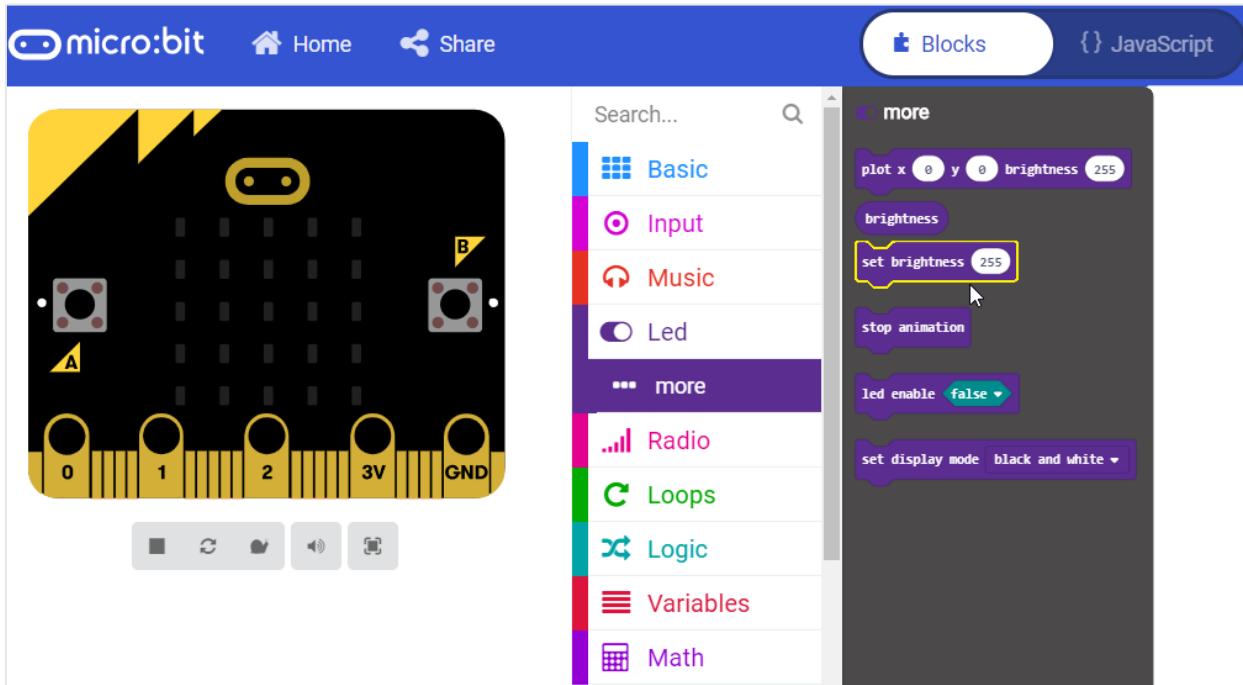
Kao argument ove naredbe na mjesto nule smjestite naredbu *analog read pin P0*. Time će se na zaslonu micro:bita ispisivati vrijednost analognog signala na izvodu P0.



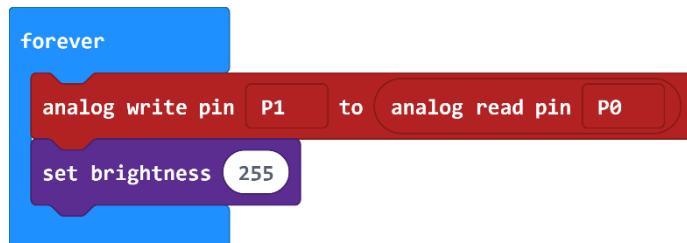
Isprobajte program na spoju. Kad je potenciometar u potpunosti okrenut prema minusu, svjetleća dioda je ugašena, a na zaslonu micro:bita se ispisuje broj 1. Kako potenciometar zakrećete prema plusu, svjetleća dioda sve jače svijetli, a na zaslonu micro:bita se ispisuju sve veći brojevi. Kad je potenciometar skroz okrenut prema plusu, svjetleća dioda svijetli najjačim intenzitetom, a na zaslonu micro:bita se ispisuje vrijednost 1023.

Modificirajte prethodni program tako da potenciometar regulira i način rada svjetlećih dioda na zaslonu micro:bita jednako kao i način rada vanjske svjetleće diode. Najprije obrišite naredbu za prikaz vrijednosti analognog signala na izvodu P0.

Za reguliranje intenziteta svjetla dioda na zaslonu micro:bita potrebna vam je naredba *set brightness* iz kategorije **Led**.



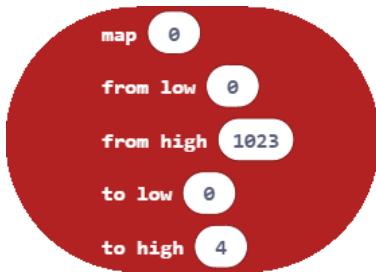
Odaberite tu naredbu i stavite je u *forever* petlju.



Trenutna vrijednost koja stoji u *set brightness* naredbi je 255 i to je maksimalna vrijednost kojom svjetleće diode na micro:bitu mogu svijetliti.

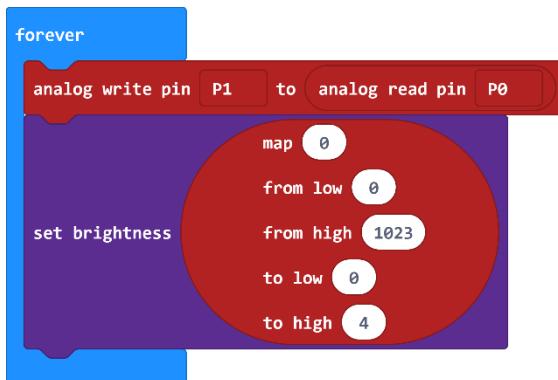
Sada je potrebno povezati vrijednost intenziteta svjetla svjetlećih dioda na zaslonu micro:bita, čiji je raspon od 0 do 255, s vrijednosti analognog signala izvoda P0 na kojem se nalazi potenciometar i čija se vrijednost kreće od 0 do 1023.

Da biste mogli povezati te vrijednosti, upotrijebite naredbu za mapiranje vrijednosti iz kategorije **Pins**.



Ovu naredbu smjestite kao vrijednost intenziteta svjetlećih dioda micro:bita u naredbu *set brightness*.

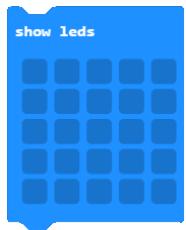
Ona omogućuje usklađivanje vrijednosti različitih raspona. Raspon vrijednosti analognog signala na izvodu P0, koji se kreće od 0 do 1023, preslikat će u raspon od 0 do 255.



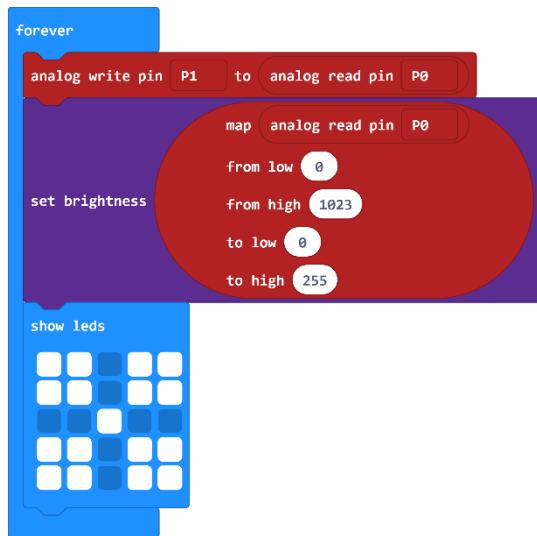
S obzirom na to da je potrebno premapirati vrijednost analognog signala izvoda P0, naredbu *analog read pin P0* smjestite u *map* dio naredbe. U *form low* ostavite 0 te u *from high* ostavite 1023 jer je to njegov raspon. U dijelu *to low* također ostavite vrijednost 0 jer je ona u novom rasponu najniža, a u dijelu *to high* upišite 255 jer je to najveća vrijednost.



Za kraj je još potrebno dodati što će se prikazati na micro:bitu. Iz kategorije **Basic** odaberite naredbu *show leds*.



Naredbu *show leds* smjestite u *forever* petlu i iscrtajte sličicu po želji.

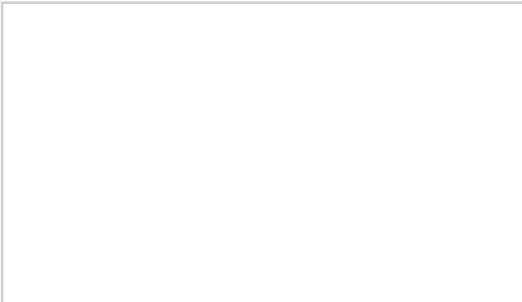


S malom izmjenom u kôdu, napravite program kojim potenciometar upravlja radom svjetlećih dioda tako da kad vanjska dioda svijetli jačim intenzitetom da diode na micro:bitu svijetle slabijim. Za to će u naredbi za mapiranje biti potrebno samo izmijeniti dio *low to* i *high to*. U *low to* upišite 255, a *high to* upišite 0. Na taj način ste napravili novo mapiranje raspona u kojem su visoke vrijednosti analognog signala povezane s niskim vrijednostima novog raspona i obrnuto.

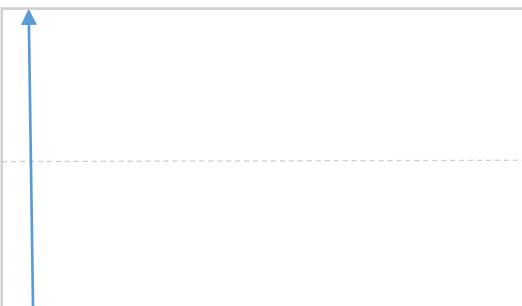
Kreativni kutak

U kreativnom kutku pogledajte kako od ovih modula i programa izraditi lampu kojoj možete regulirati način na koji će svjetliti. Od kolaža i ljepila izradite lampu prema uputama.

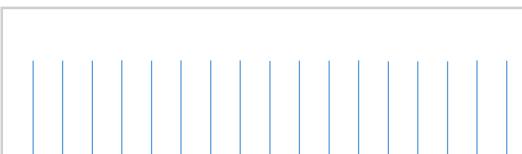
Postavite kolaž papir u vodoravni položaj.



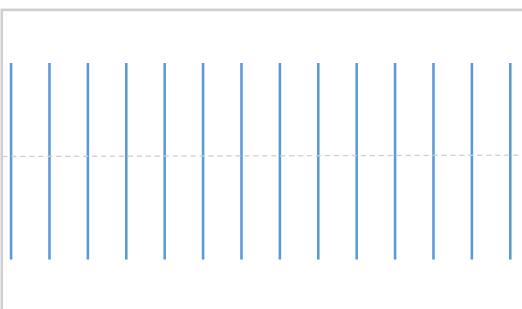
Presavnite ga odozdo prema gore.



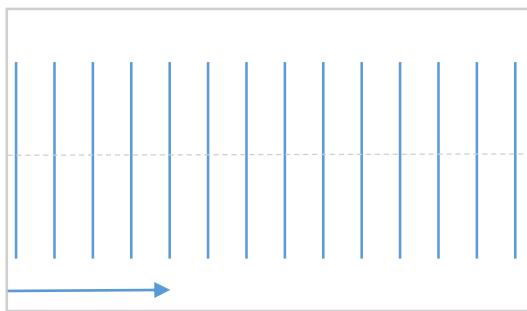
Iscrtajte ravne linije od sredine presavinutog papira prema gore (ali ne kraja).



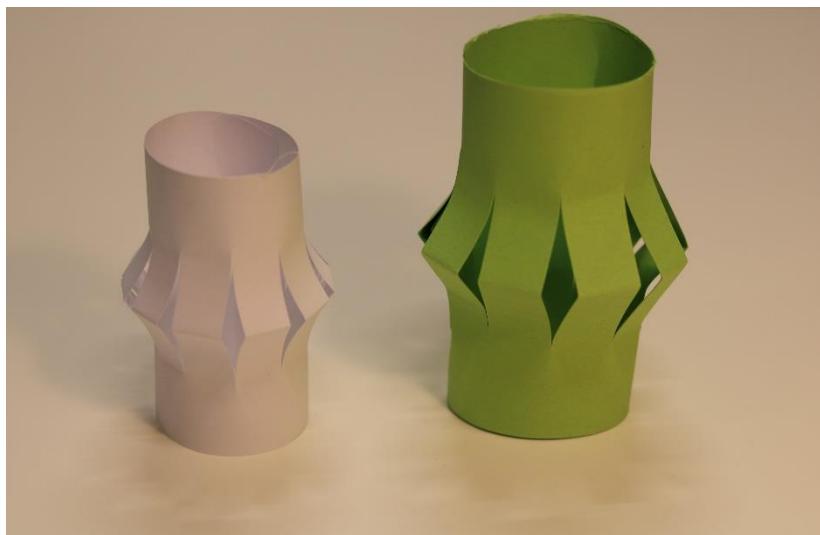
Po iscrtanim linijama izražite papir i nakon toga ga rastvorite.



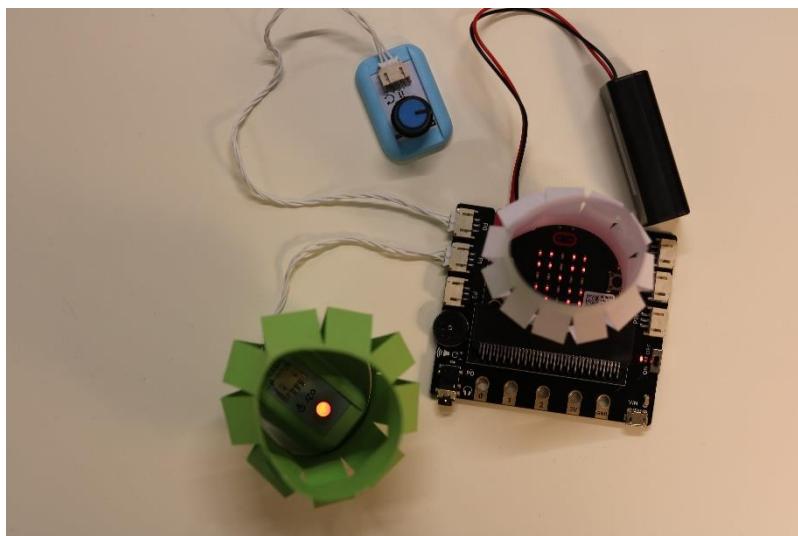
Zarolajte papir u jednu udesno i zaljepite na rubovima.



Primjeri origami lampi:



Moduli kao lampa:



Projekt 3: Ventilator

Opis projekta

U ovom projektu upoznat ćete rad ventilatora koji kao izlazni modul seta može biti digitalni ili analogni.

U prvom primjeru povezat ćete u spoj potenciometar čiji ste rad upoznali u prethodnom primjeru.

Okretanjem potenciometra regulirat ćete brzinu okretanja ventilatora. U tom slučaju će ventilator biti analogni izlaz. Izradite program tako da se ventilator okreće trima različitim brzinama ovisno o položaju potenciometra.

U drugom primjeru ćete kao ulazni element upotrijebiti temperaturni senzor micro:bita. Ako je temperatura veća od 25 stupnjeva, ventilator će krenuti s radom najjačom brzinom. Kad je temperatura manja od 25 stupnjeva, ventilator neće raditi. U tom slučaju će se ventilator ponašati kao digitalni izlazni modul. Neka se na zaslonu micro:bita ispisuje vrijednost temperature u stupnjevima Celzijevim svakih 3 sekunde.

Pomoću ovih modula i programa u kreativnom dijelu projekta možete poput nas izraditi skladište u kojem ventilator radi kada je temperatura previsoka ili osmislite svoju kreativnu priču i odaberite materijale kojima ćete ventilator i potenciometar pretvoriti u funkcionalni sustav.

Potrebni moduli

Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 2 kabela za spajanje modula, potenciometar, ventilator

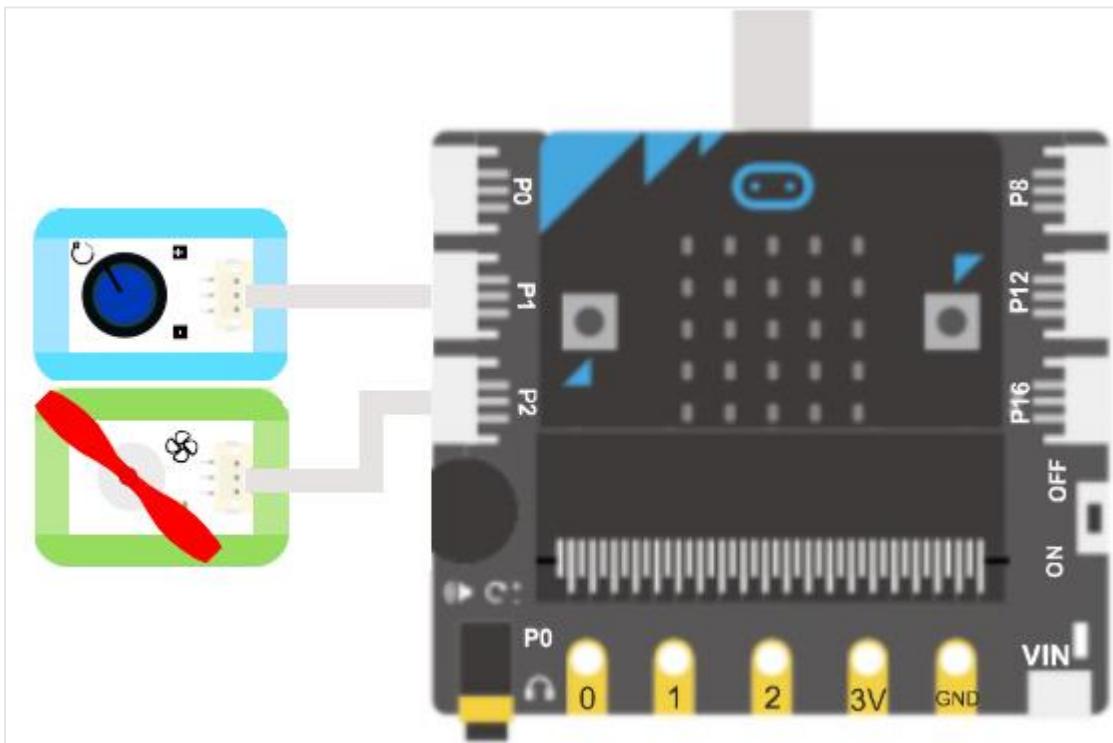
Spajanje modula

U pločicu smjestite micro:bit. Na mjestima za spajanje elektroničkih modula kablovima povežite i spojite potenciometar te ventilator kao što je prikazno na montažnoj shemi. Naučili ste u prethodnom projektu da se potenciometar kao analogni ulazni modul može spojiti samo na izvode koji su analogni. To su izvodi P0, P1 ili P2 na pločici. Micro:bit preko izvoda na kojem je spojen potenciometar kao analogni ulazni modul, očitava stanje njegovog signala čiji se raspon kreće u vrijednostima od 0 do 1023. Ventilator može biti digitalni ili analogni izlazni modul. U ovom projektu ćete u prvom primjeru koristiti ventilator kao analogni modul, a u drugom kao digitalni. S obzirom na to ventilator možete spojiti također samo na izvode P0, P1 ili P2. Ovisno o vrijednosti analognog signala koja će biti postavljena na izvod ventilatora, on će se okretati brže ili sporije. Prema prikazu spajanja, potenciometar spojite na izvod P1, a ventilator

na izvod P2. USB kabelom povežite micro:bit s računalom kako bi spoj dobio napajanje i kako biste prebacili program na micro:bit.

Napomena: USB kabel će nakon prijenosa programa biti potrebno spojiti na VIN USB priključak koji omogućava napajanje modula kojima je potrebno više električne energije za rad.

Prikaz spajanja



Izrada programa

Potrebne naredbe: *forever, analog write pin P0 to 1023, analog read pin P0, show number, if then else, operator za usporedbu veličina, set variable to*

Sve naredbe i u ovom programu smjestite u *forever* petlju. *Forever* petlja nalazi se u kategoriji **Basic**. Na taj način moći ćete cijelo vrijeme regulirati brzinu okretanja ventilatora.



U prvom primjeru će ventilator biti analogni izlazni modul. Micro:bit će stoga na njegovom izvodu postavljati vrijednost koja se kreće u rasponu od 0 do 1023. U *forever* petlju postavite naredbu *analog write pin P0 to 1023* iz kategorije **Pins**. Promijenite parametar P0 u P2 jer je ventilator spojen na izvod P2.



Kao i u prethodnom projektu, vrijednost signala izvoda potenciometra koju micro:bit očitava, on postavlja kao vrijednost signala izvoda ventilatora. Iz kategorije **Pins** odaberite naredbu *analog read pin P0* i promijenite parametar P0 u P1 jer je na izvodu P1 spojen potenciometar. Tu naredbu postavite kao argument naredbe *analog write pin P0*.



Prebacite program na micro:bit i isprobajte ga.

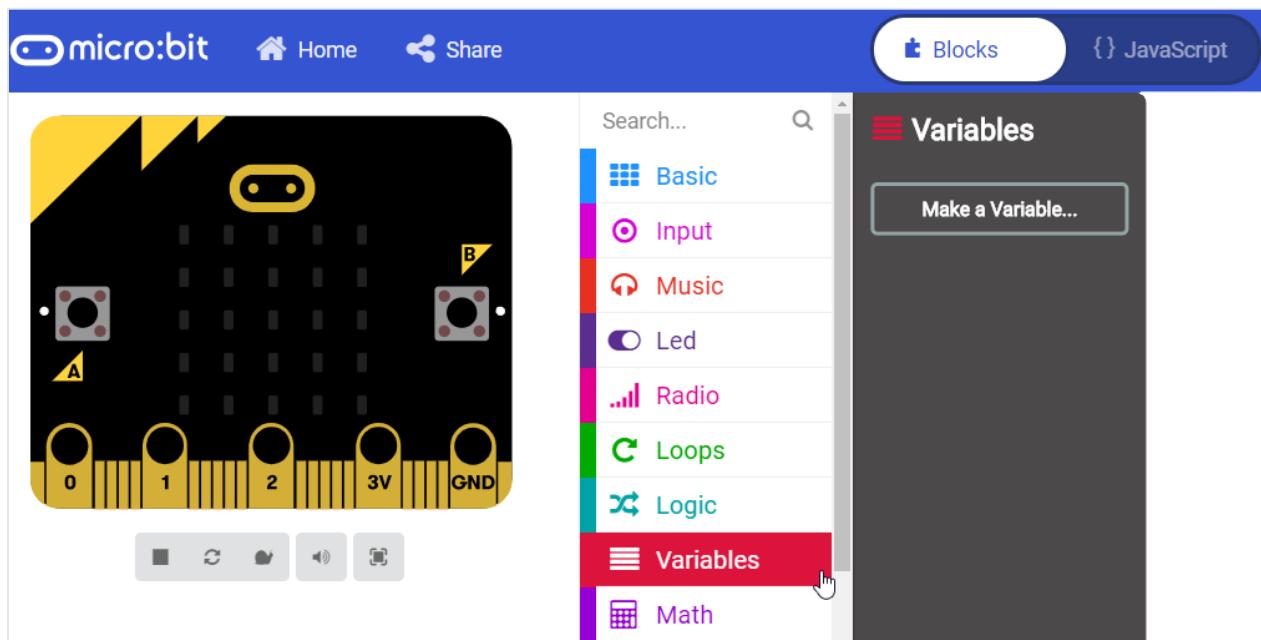
Primijetite da je za pokretanje ventilatora potrebna veća snaga, stoga potenciometar zakrećite dok se ventilator ne pokrene.

Podsjetnik: USB kabel će nakon prijenosa programa biti potrebno spojiti na VIN USB priključak koji omogućava napajanje modula kojima je potrebno više električne energije za rad.

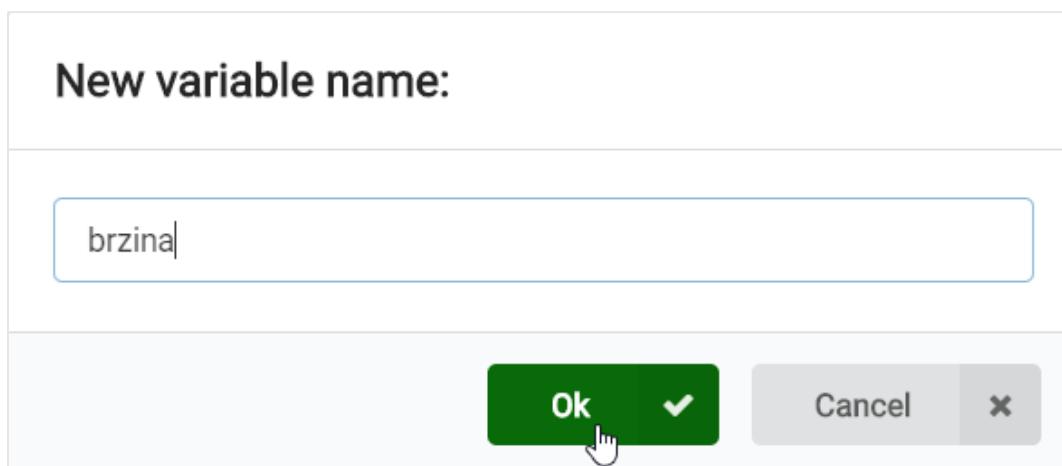
Modificirajte sada program tako da se ventilator okreće trima različitim brzinama ovisno o tome u kojem je položaju potenciometar.

Za to će biti potrebno kreirati varijablu u koju će se pohranjivati analogna vrijednost potenciometra.

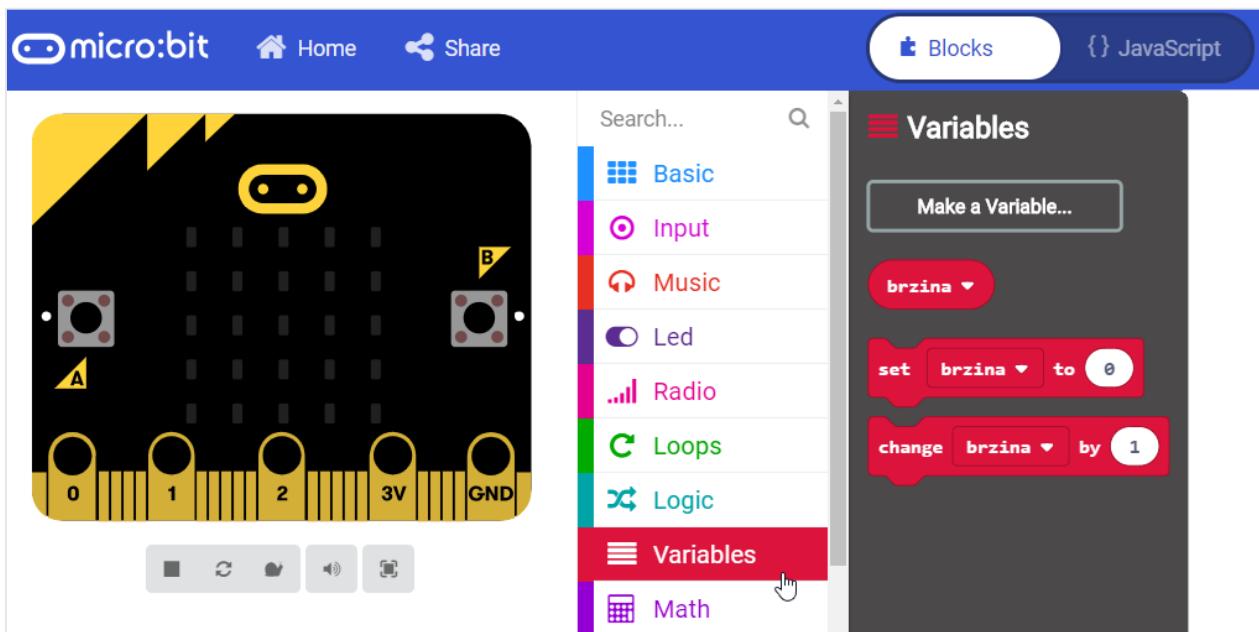
Za kreiranje varijable odaberite kategoriju **Variables**.



Odaberite opciju *Make a Variable...* i u prozoru koji se otvoru upišite ime varijable. Neka se varijabla zove *brzina*. Pritisnite na *Ok*.



Kreiranjem varijable u kategoriji Variables pojavljuju se tri naredbe. Prva naredba, *brzina*, je vrijednost varijable. Druga naredba *set brzina to 0* služi za inicijaliziranje varijable u programu i postavljanje njezine vrijednosti. Naredba *change brzina by 1* služi za mijenjanje vrijednosti varijable (povećavanje ili smanjivanje).



U *forever* petlji postavite naredbu *set brzina to 0*.



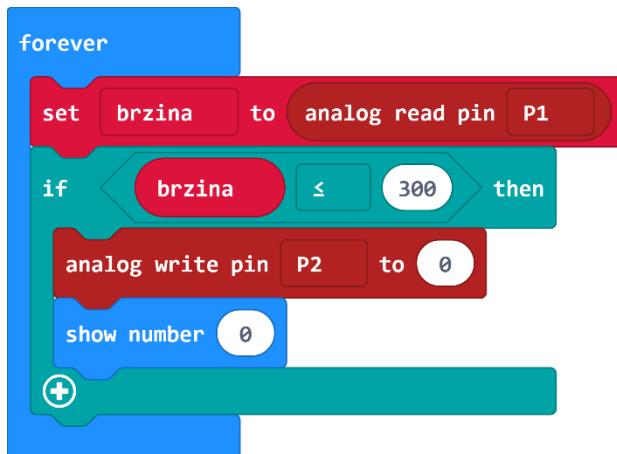
Potrebno je umjesto broja 0 kao vrijednost varijable brzina postaviti naredbu *analog read pin P1* iz kategorije **Pins** jer je na izvodu P1 spojen potenciometar. Na taj način u svakom trenutku izvršenja programa varijabla brzina ima vrijednost koja odgovara analognoj vrijednosti izvoda P1 koju micro:bit čita.



Sada slijedi izrada dijelova programa u kojima ćete definirati da se ventilator okreće trima različitim brzinama ovisno o vrijednosti varijable *brzina*.

Najprije postavite uvjet kojim se ventilator neće okretati ako je vrijednost varijable *brzina* manja ili jednaka 300. To znači da kad je potenciometar postavljen položaj od samog početka s lijeve strane do mesta gdje daje analognu vrijednost 300, ventilator se neće pokrenuti.

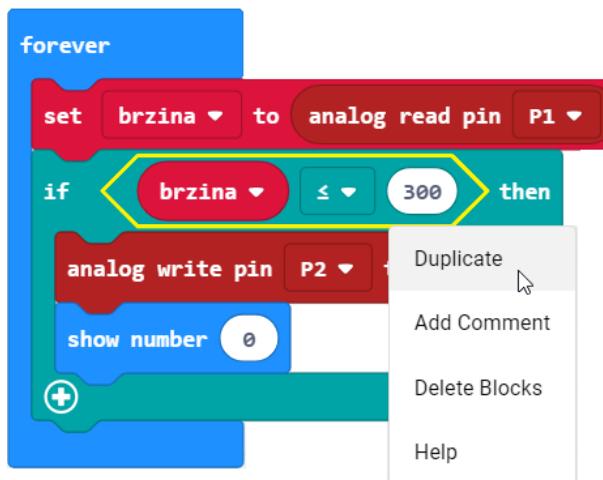
Iz kategorije **Logic** odaberite *if then* naredbu i smjestite je u *forever* petlju. U *if* dio smjestite operator usporedbe i postavite ga na opciju manje ili jednako (\leq). S lijeve strane dodajte vrijednost varijable *brzina* (prva naredba iz kategorije **Variables**), a s desne strane upišite broj 300. U *then* dio naredbe postavite naredbu *analog write pin P2 to 0* iz kategorije **Pins** te dodajte naredbu *show number 0* kako bi se na zaslonu micro:bita prikazao broj 0 kad ventilator ne radi.



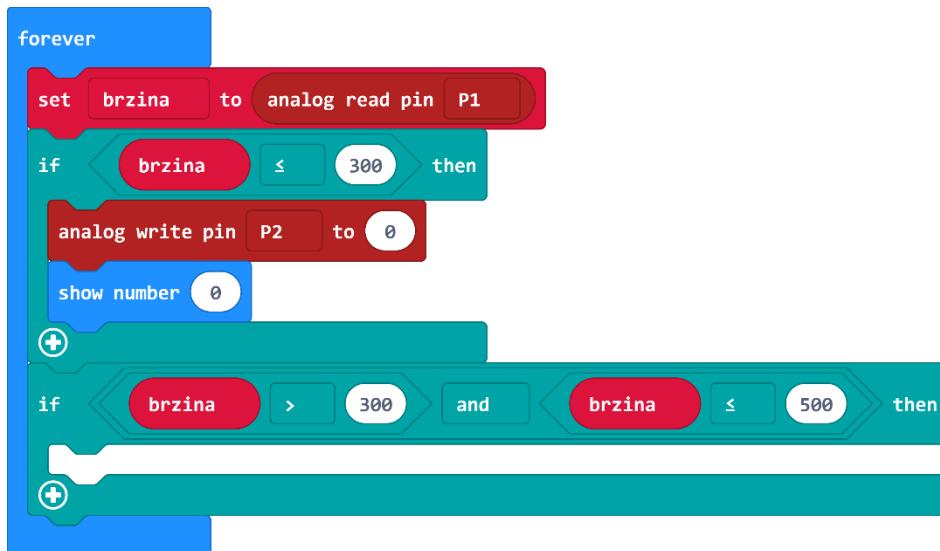
Sada ćete definirati uvjet za okretanje ventilatora prvom brzinom. Ako je vrijednost varijable veća od 300 i manje ili jednaka 500 tada neka se ventilator okreće brzinom 600. Ponovno ćete upotrijebiti naredbu *if then* iz kategorije **Logic**, no sada vam je potrebna i logička naredba *and* kojom zadati više uvjeta istovremeno. Ona se nalazi u kategoriji **Logic**. Smjestite ju u *if* dio nove *if then* naredbe.



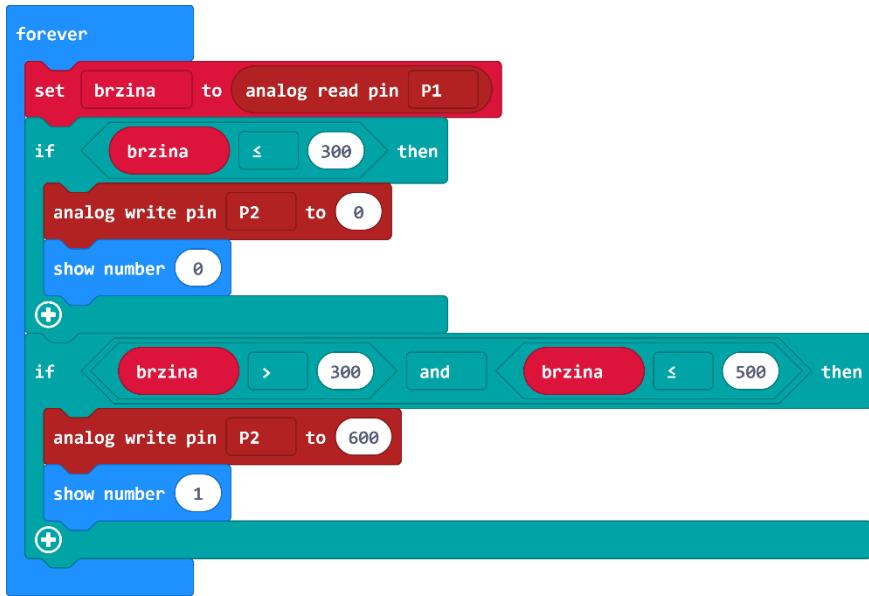
U prvom dijelu and naredbe postavite uvjet da je brzina veća od 300, a u drugom dijelu da je manja ili jednaka 500. Najjednostavniji način za to jest da 2 puta kopirate naredbu usporedbe iz prve *if then* naredbe. Zatim kopirane naredbe postavite na predviđena mjesta u novoj *if then* naredbi te promijenite vrijednosti. Pritisnite na naredbu usporedbe i zatim desni klik miša te odaberite opciju Duplicate. To ponovite još jednom.



Kopirane naredbe smjestite u logičku naredbu and i izmjenite vrijednosti i operatore da odgovaraju novom uvjetu (brzina veća od 300 i manje ili jednaka 500).

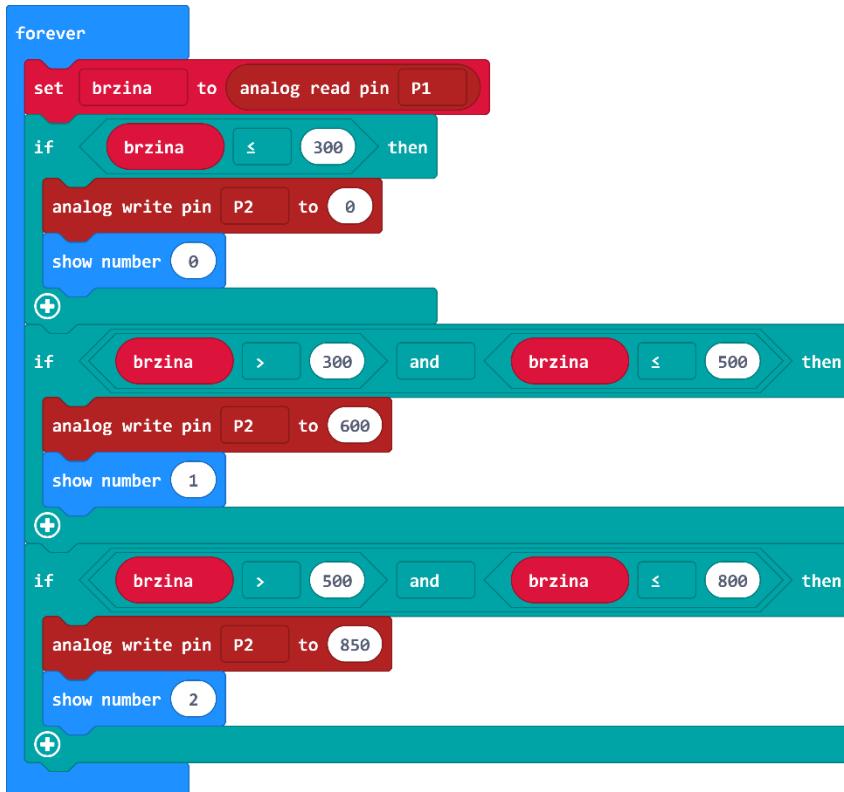


U then dio naredbe postavite naredbu *analog write pin P2 to* i postavite vrijednost 600 kao argument naredbe. Dodajte još naredbu *show number 1* kako bi se na zaslonu micro:bita prikazao broj 1 kad ventilator radi prvom brzinom.



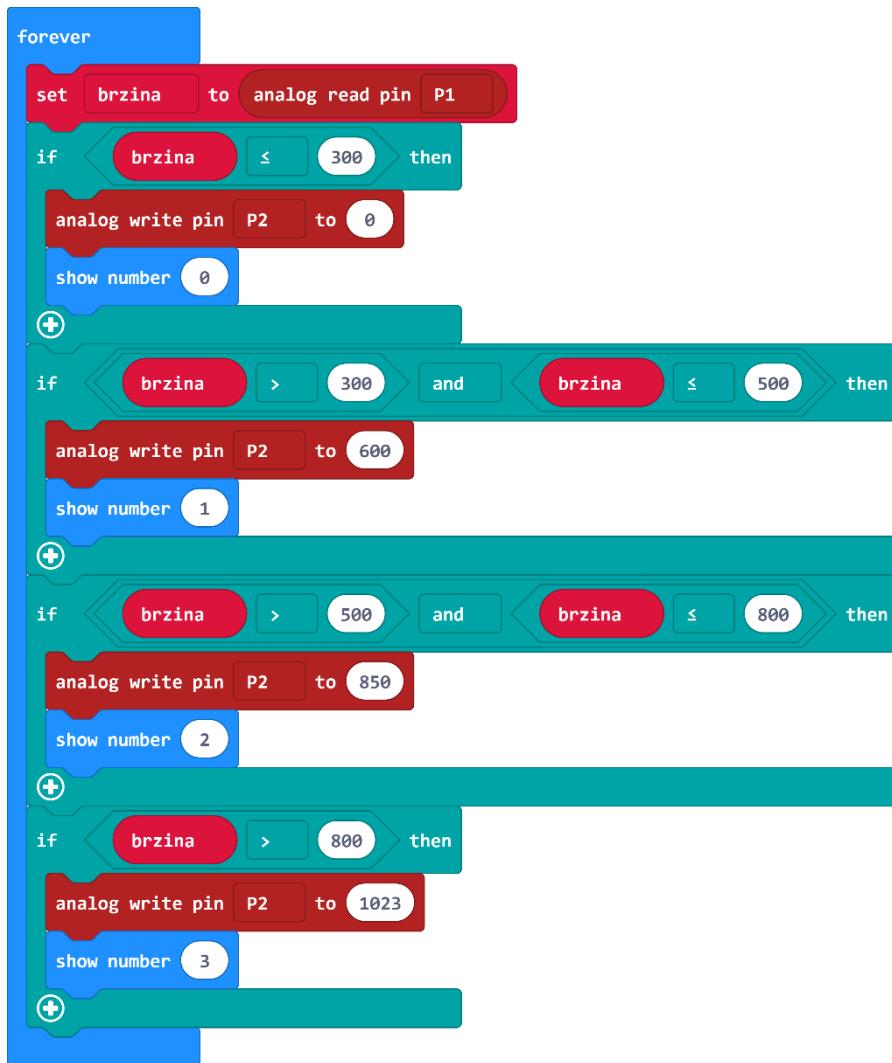
```
forever
  set brzina to (analog read pin P1)
  if (brzina <= 300) then
    analog write pin P2 to 0
    show number (0)
  end
  if (brzina > 300 and brzina <= 500) then
    analog write pin P2 to 600
    show number (1)
  end
```

Za drugu brzinu kopirajte cijelu posljednju *if then* naredbu i kopirane blokove postavite na kraj u *forever* petlji. Napravite izmjene vrijednosti tako da novi uvjet glasi: ako je brzina između 500 i 800 neka se na izvod potenciometra postavi vrijednost 850. Time će se ventilator okretati brže kad se potenciometar zaokrene u desno do položaja na kojem ima analognu vrijednost 800.



```
forever
  set brzina to (analog read pin P1)
  if (brzina <= 300) then
    analog write pin P2 to 0
    show number (0)
  end
  if (brzina > 300 and brzina <= 500) then
    analog write pin P2 to 600
    show number (1)
  end
  if (brzina > 500 and brzina <= 800) then
    analog write pin P2 to 850
    show number (2)
  end
```

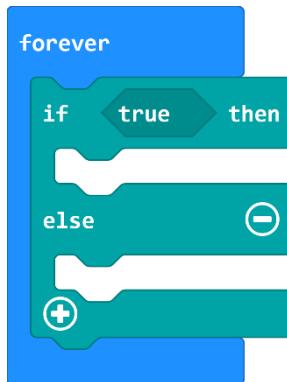
Za treću brzinu ponovite kopirajte prvu if then naredbu i postavite ju na kraj u forever petlji. Napravite izmjene tako da novim uvjetom ventilator radi punom brzinom (analog write pin P2 to 1023) ako je vrijednost brzine veća od 800.



U drugom primjeru programa ventilator će biti digitalni izlazni modul koji će se pokrenuti kad temperatura bude veća od 25 stupnjeva.

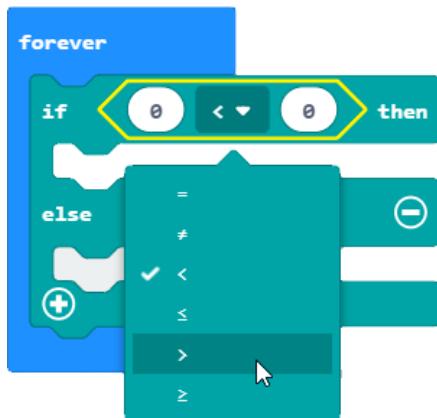
Potrebne naredbe: *forever, if then else, temperature (°C), naredba za usporedbu, show number, show string, digital write pin P0 to 0, pause (ms)*

U petlju forever postavite naredbu *if then else* iz kategorije **Logic**.

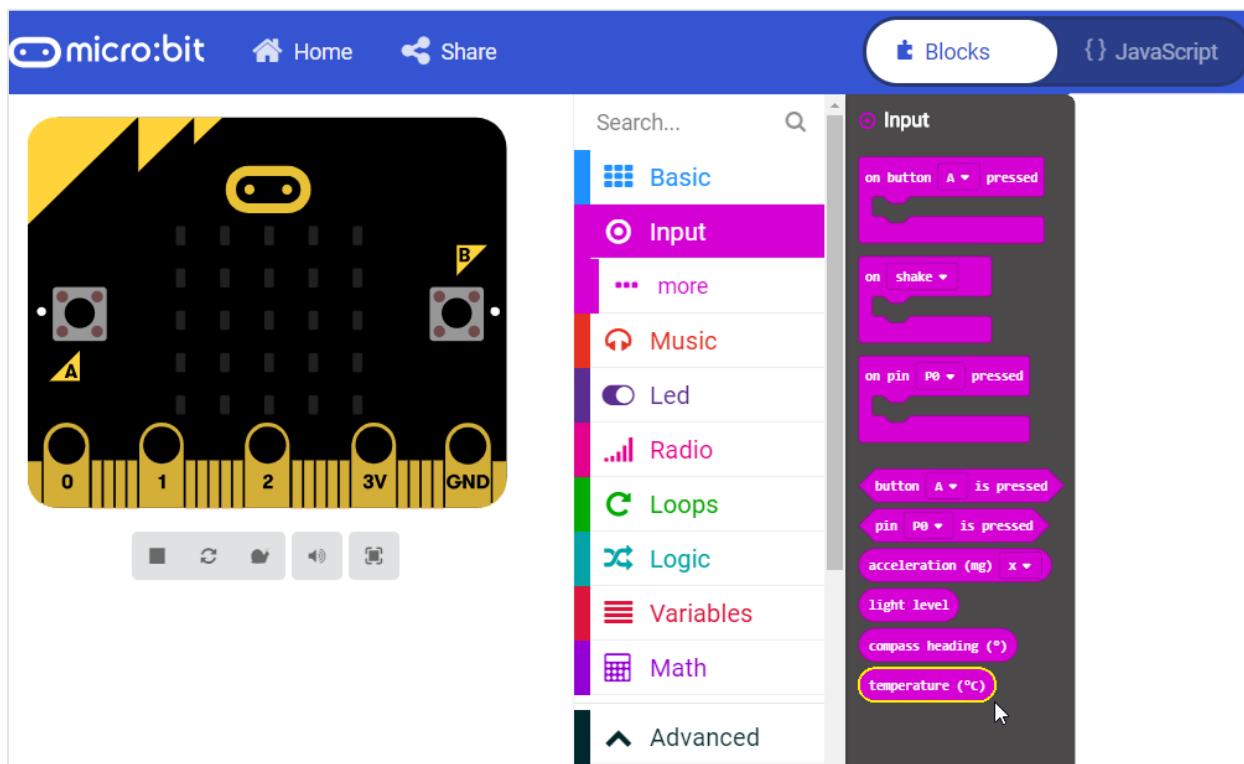


U if dijelu naredbe definirat ćete uvjet za pokretanje rada ventilatora. Uvjet glasi: ako je temperatura veća od 25.

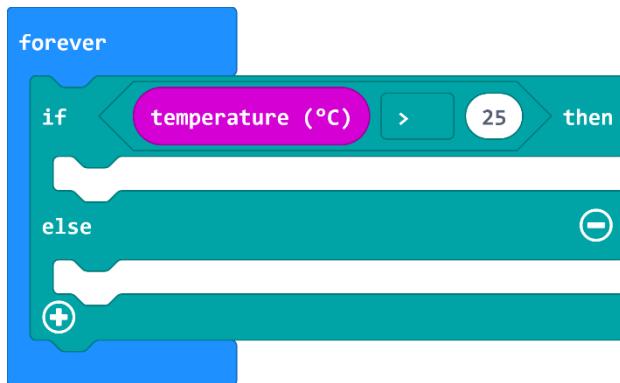
Iz kategorije Logic odaberite naredbu za usporedbu i postavite ju u *if* dio naredbe. Potrebno je još promijeniti znak jednakosti u znak veće. To ćete učiniti tako da pritisnete na strelicu uz znak jednakosti i s popisa koji se otvorí odaberete znak usporedbe veće.



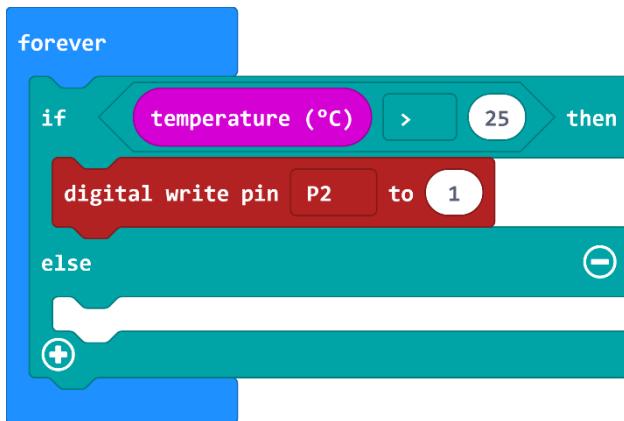
Iz kategorije **Input** odaberite naredbu *temperature (°C)*.



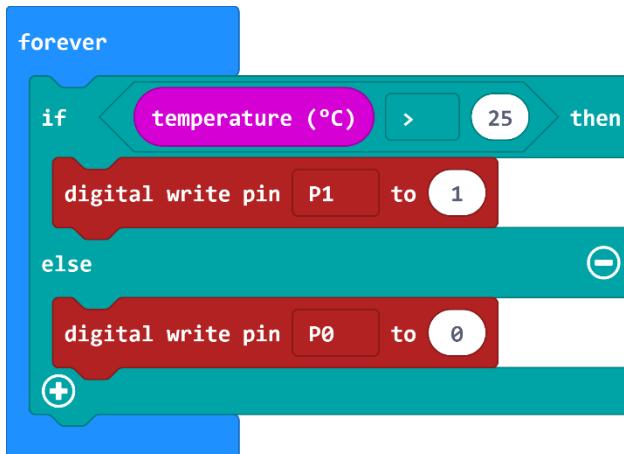
Naredbu *temperature (°C)* smjestite s lijeve strane usporedbe, a s desne upišite broj 25. Time je definiran uvjet, a sada treba u programskom kôdu dodati naredbu koja će se izvršiti kad je taj uvjet istinit.



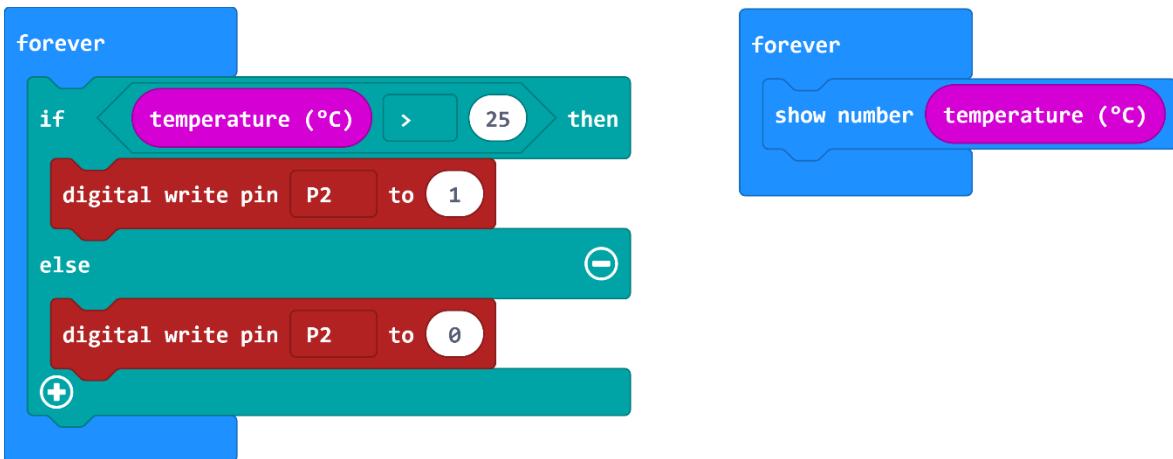
Iz kategorije **Pins** odaberite naredbu *digital write pin P0 to 0* i smjestite ju u *then* dio. Promijenite parametar P0 u P2 jer je ventilator spojen na tom izvodu. Kao argument te naredbe umjesto nule upišite 1. Sada u programskom kôdu stoji da će ventilator početi s radom kada stanje izvod P2 postane 1, a to se dogodi kad je temperatura veća od 25 °C.



Inače, ako je temperatura manja ili jednaka 25, ventilator neka ne radi. Zato iz kategorije Pins u *else* dio postavite naredbu *digital write pin P0 to 0* i umjesto P0 postavite P2 te ostavite vrijednost argumenta 0.



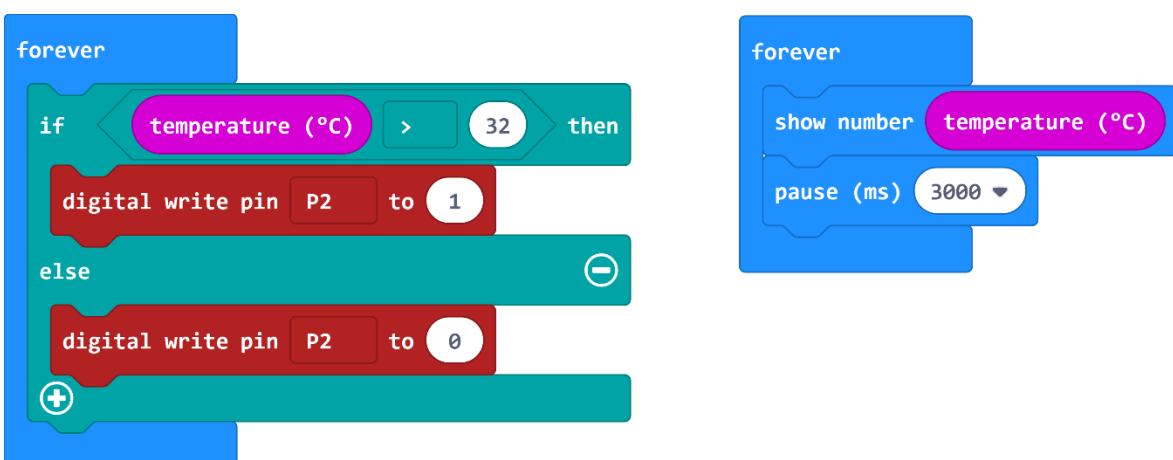
Da biste lakše pratili vrijednost temperature neka se na zaslonu micro:bit ispisuje njezina vrijednost. Iz kategorije **Basic** odaberite novu *forever* petlju i smjestite ju u prostor za programiranje. U nju iz kategorije **Basic** smjestite naredbu *show number*. Kao argument te naredbe postavite naredbu *temperature (°C)* iz kategorije **Input**.



Da bi se temperatura ispisivala svakih 3 sekunde potrebno je ispod ispisa njezine vrijednosti dodati *naredbu za pauzu*. Ta se naredba nalazi u kategoriji **Basic**.



Upišite na mjestu argumenta 3000 jer je to 3 sekunde (3000 milisekundi).



Prebacite program na micro:bit i isprobajte ga.

Kreativni kutak

Pomoću ovih modula i programa u kreativnom dijelu projekta možete poput nas izraditi skladište u kojem ventilator radi kada je temperatura previsoka ili osmislite svoju kreativnu priču i odaberite materijale kojima će ventilator i potenciometar pretvoriti u funkcionalni sustav.



Projekt 4: Mjerač zvuka

Opis projekta

U ovom projektu upoznat ćete senzor zvuka koji je analogni ulazni modul. Pomoću tog senzora možete detektirati glasnoću zvukova u prostoru. S obzirom na to izradite mjerač zvučnog zagadjenja prostora uz pomoć servo motora koji će na skali prikazati koliko je vam je glasna okolina u kojoj se nalazite. Neka postoje dvije razine glasnoće – tiho i glasno. Testirajte program i postavite graničnu vrijednost prema glasnoći u vašem prostoru.

Uz manje izmjene u programu mjerač pretvorite u uređaj za društvenu igru u kojoj skupljate bodove ako dovoljno tiho obavite zadatku. Možete učiniti i korak dalje te uz pomoć istih modula začiniti rođendansku zabavu na kojoj ćete glasnim pjevanjem otvoriti pretinac iz kojeg će ispasti slatkiši.

Iz kreativnog kutka odaberite model dijagrama za određivanje razine zvuka.

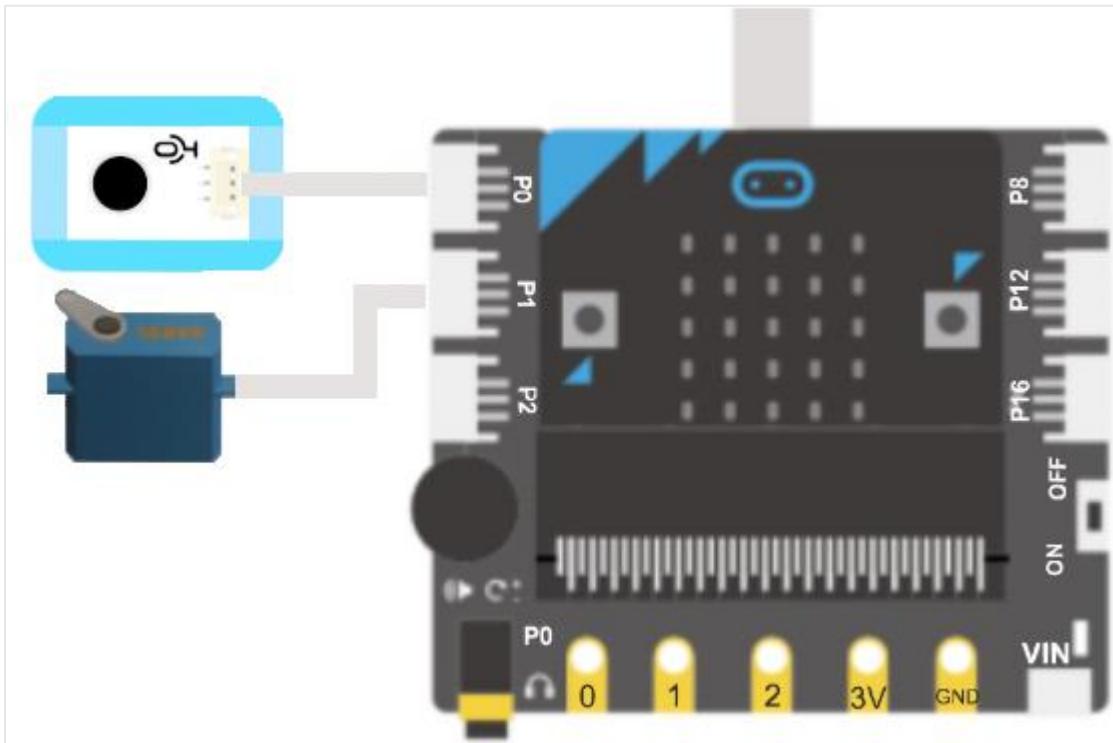
Potrebni moduli

Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 2 kabela za spajanje modula, senzor za zvuk, servo motor

Spajanje modula

Senzor za zvuk je analogni ulazni modul koji se može spojiti samo na izvode P0, P1 ili P2. U ovom primjeru, on je spojen na izvod P0, a servo motor na izvod P1. Prema prikazu spajanja kabelima povežite module na izvode upravljačke pločice. Možete ih rasporediti i na druga mesta ako vam tako bolje odgovara kad ćete spoj implementirati u svoj kreativni rad. Pripazite pri tome da i u programskom kôdu budu postavljeni odgovarajući izvodi jer u suprotnome se program neće izvršavati.

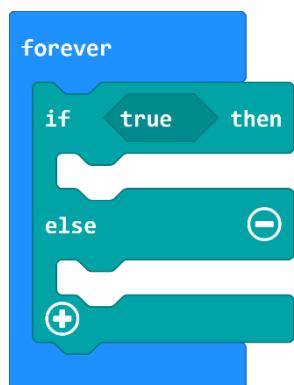
Prikaz spajanja



Izrada programa

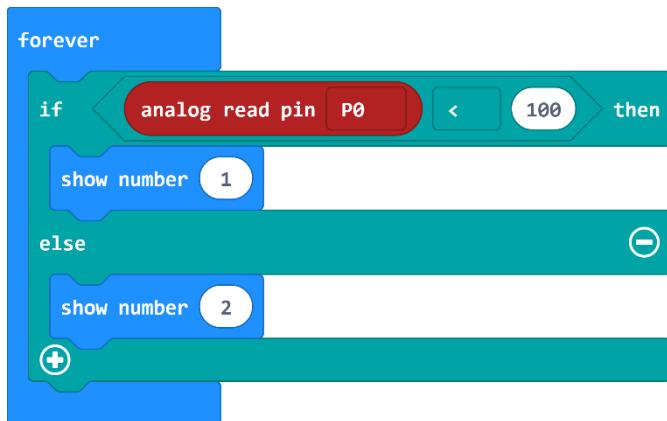
Potrebne naredbe: *forever, if then else, analog red pin P0, operatori za usporedbu veličina, show number, servo write pin P0*

Za početak u *forever* petlju postavite logičku naredbu *if then else*.



U logičkoj naredbi ćete definirati u kojem položaju mora biti servo motor ovisno o tome koju granicu glasnoće zvuka postavite. Najprije testirajte senzor prema postavljenoj granici i uvjetima u kojima se

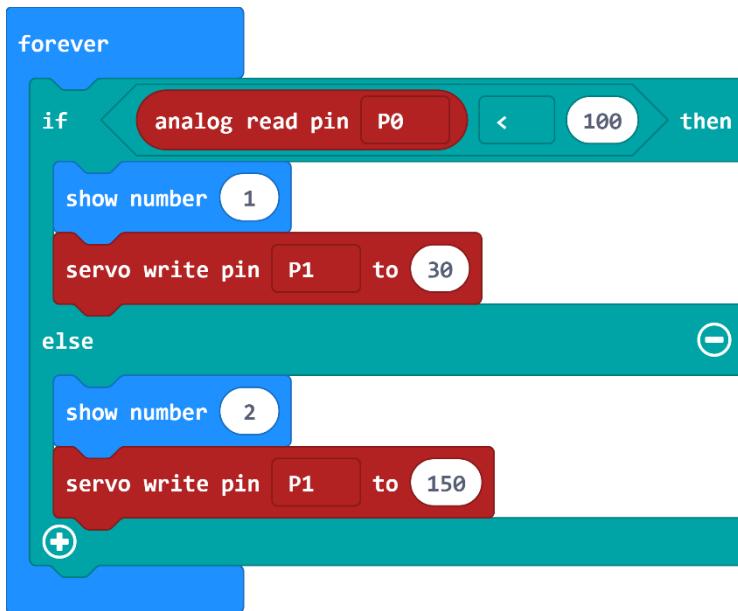
nalazite tako da se na zaslonu micro:bita ispiše broj 1 ako je glasnoća ispod granice, odnosno 2 ako je glasnoća iznad granice. Za ispis brojeva koristite *show number* naredbu iz kategorije **Basic**. Za postavljanje logičkog uvjeta upotrijebite *naredbu za usporedbu manje* iz kategorije **Logic** te postavite naredbu *analog read pin P0* s lijeve strane znaka manje od, a s desne strane upišite vrijednost 100.



Prebacite program na micro:bit i testirajte ga mijenjajući glasnoću u prostoru te pratite promjene i uočite povezanost između glasnoće zvuka i brojeva koji se prikazuju. Ako ste zadovoljni možete ostaviti granicu na 100, a ako vam bolje odgovara neka druga vrijednost, upišite nju.

Sada ćete definirati položaj u koji se mora zaokrenuti servo motor ovisno o glasnoći zvuka. Kad je glasnoća manja od postavljene granice servo motor neka bude u jednom položaju (primjerice lijevo), a kad je glasnoća prijeđe granicu, neka se servo motor zaokrene udesno u drugi položaj.

Za pokretanje servo motora upotrijebite naredbu *servo write pin P0* te promijenite parametar P0 u P1 jer je servo motor spojen na izvod P1. Naredbu duplicitirajte (desni klik na naredbu i pritisnite na *Duplicate*). Jednu smjestite u *then* dio naredbe ispod ispisa broja, a drugu u *else* dio. Upišite vrijednosti za zaokretanje servo motora u obje naredbe. U prvoj neka vrijednost bude 30, a u drugoj 150. Možete unijeti i neke druge vrijednosti ako vam tako bolje odgovara radi kreativnih dodataka u spoju (Pogledajte naš primjer i prema njemu izradite svoj).



Završili ste program za mjerač zvuka. U kreativnom kutku ovog projekta potražite dodatak kojim ćete personalizirati svoj spoj modula.

Ako vam se svidio ovaj primjer, možete učiniti promjene u programu i pretvoriti spoj u uređaj za društvenu igru. Cilj igre je obaviti zadatku dovoljno tiho u zadanom vremenu.

Igra kreće odmah kad pokrenete micro:bit i traje 30 sekundi.

Za početak programa izradite kreirajte varijablu *vrijeme* i dodijelite joj vrijednost 30. Naredbu smjestite u *on start* blok. Detalje upute o kreiranju varijable pogledajte u projektu *Ventilator*.

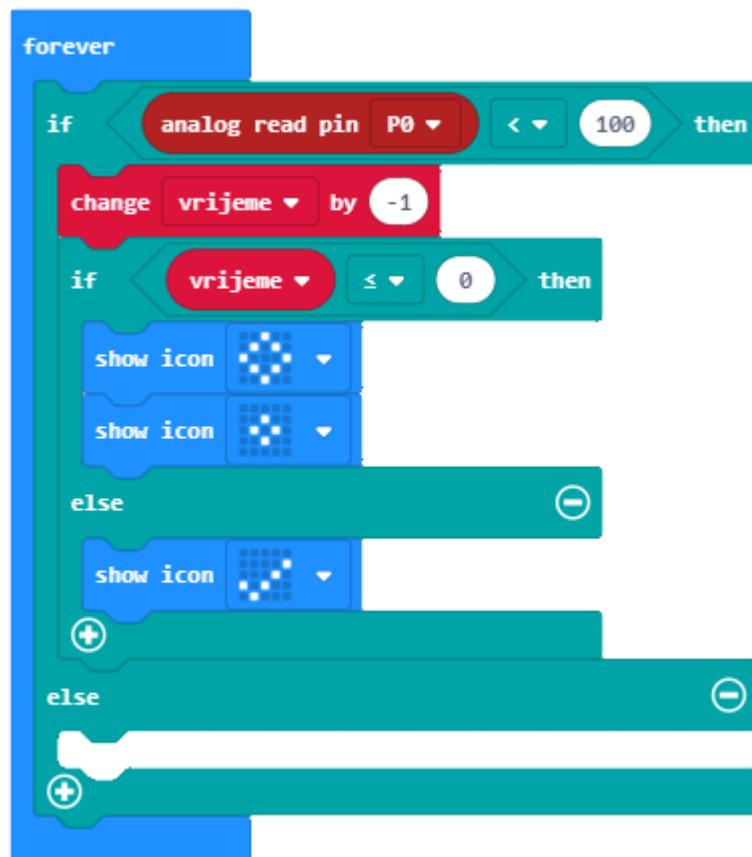


Za vrijeme igre cijelo vrijeme se prati razina zvuka. Zadajte uvjet ako je ona manja od 100 (možete mijenjati tu vrijednost po svojoj želji), neka se smanjuje vrijeme za 1.

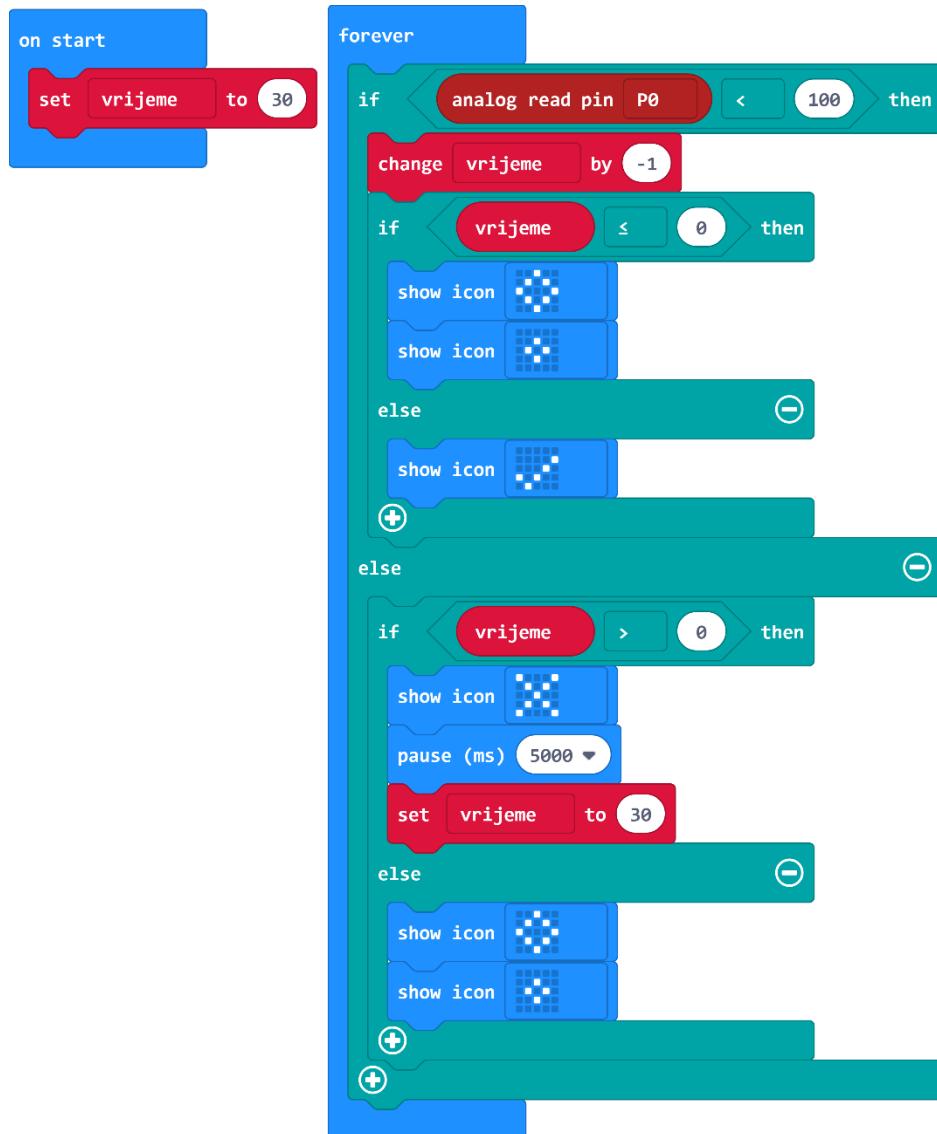
Za postavljanje uvjeta koristite *if then else* naredbu iz kategorije **Logic**. Za smanjivanje vrijednosti za 1 koristite naredbu *change vrijeme by*, te umjesto 1 upišite minus (-) 1.



Zadržite se još na *then* dijelu naredbe. Ako ste uspješno odigrali igru (to znači da je vrijeme iscurilo te je jednako ili manje od 0), neka se prikaže animacija po vašoj želji. A dok vrijeme još teče neka se na zaslonu micro:bita prikazuje kvačica. Za to upotrijebite *novu if then else* naredbu i ugnijezzdite ju u *then* dio naredbe.

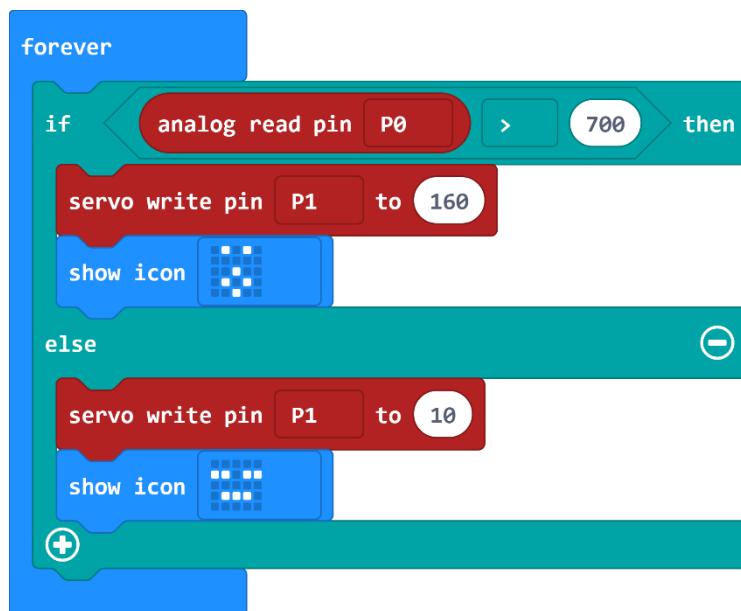


U else dijelu prve if then else naredbe treba definirati što će se dogoditi ako za vrijeme igre (dok je vrijeme veće od 0) budete preglasni. Tada neka se na zaslonu micro:bita prikaže X i nakon 5 sekundi neka igra krene ispočetka. To znači da varijabla vrijeme ponovno poprima vrijednost 30. Inače ako u trenutku kad je vrijeme iscurilo budete preglasni prije ponovnog početka igre, neka se prikazuje ista animacija kao kad vrijeme istekne.

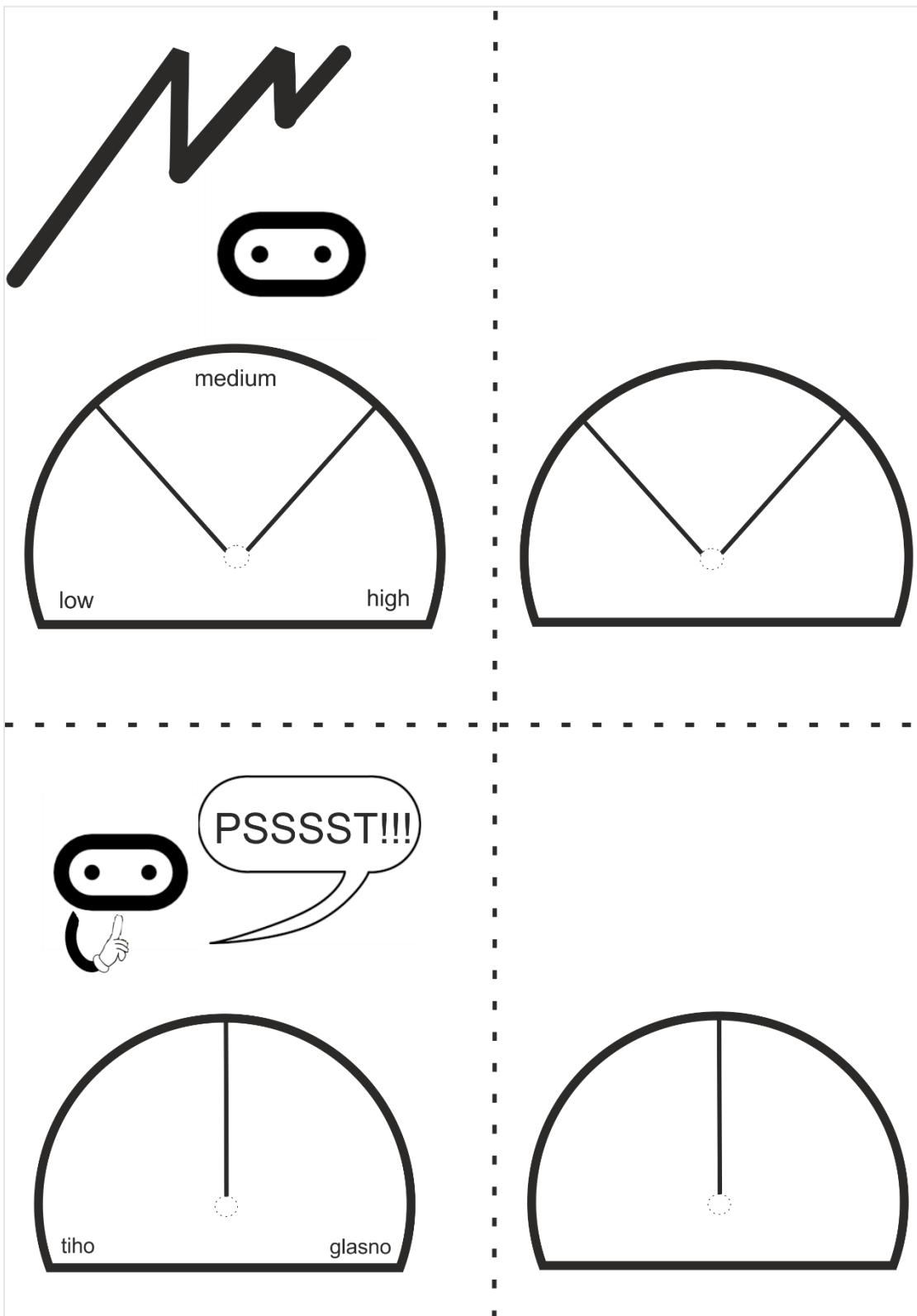


Nastavite zabavu sa senzorom za zvuk i servo motorom. Za rođendansku proslavu možete osmisliti i izraditi projekt kojim će servo motor pomaknuti polugicu i otvoriti pretinac iz kojeg će ispasti slatkiši ako s prijateljima bude pjevali dovoljno glasno.

U forever petlju postavite if then else naredbu iz kategorije Logic. Kao uvjet definirajte ako je analogna vrijednost izvoda na kojem je spojen senzor zvuka veća od 700 neka se servo motor pomakne, a na zaslonu micro:bita prikaže iznenađeno lice. Inače neka servo motor bude u položaju kojime je pretinac sa slatkisima zatvoren, a micro:bit neka spava.



Kreativni kutak



Projekt 5: Poruka za sreću

Opis projekta

U ovom projektu upoznajte rad servo motora koji je u Boson setu izlazni modul. Uz njega možete pronaći 4 polugice i vijke kojima ih možete pričvrstiti za motor. Izradite pomoću servo motora i tipkala sustav za podizanje razine sreće. Kad pritisnete tipkalo neka se pomakne polugica servo motora na kojoj se nalazi lijepa poruka ili smiješna sličica da vam uljepša dan. Spoj smjestite uz monitor ako radite na računalu ili uz knjige i bilježnice ako pišete zadaću.

Poruku ili sličicu zalijepite na polugicu. Na početku programa i sve dok tipkalo nije pritisnuto polugica je u položaju 0, okrenuta lijevo. Kad se pritisne tipkalo polugica servo motora će se okrenuti za 180 stupnjeva u desno i pokazat će vam se poruka. Kad se polugica pomakne i pokaže poruku, neka se na zaslonu micro:bita prikaže smajlić. Dok tipkalo nije pritisnuto neka se na zaslonu micro:bita slučajnim odabirom pale i gase svjetleće diode.

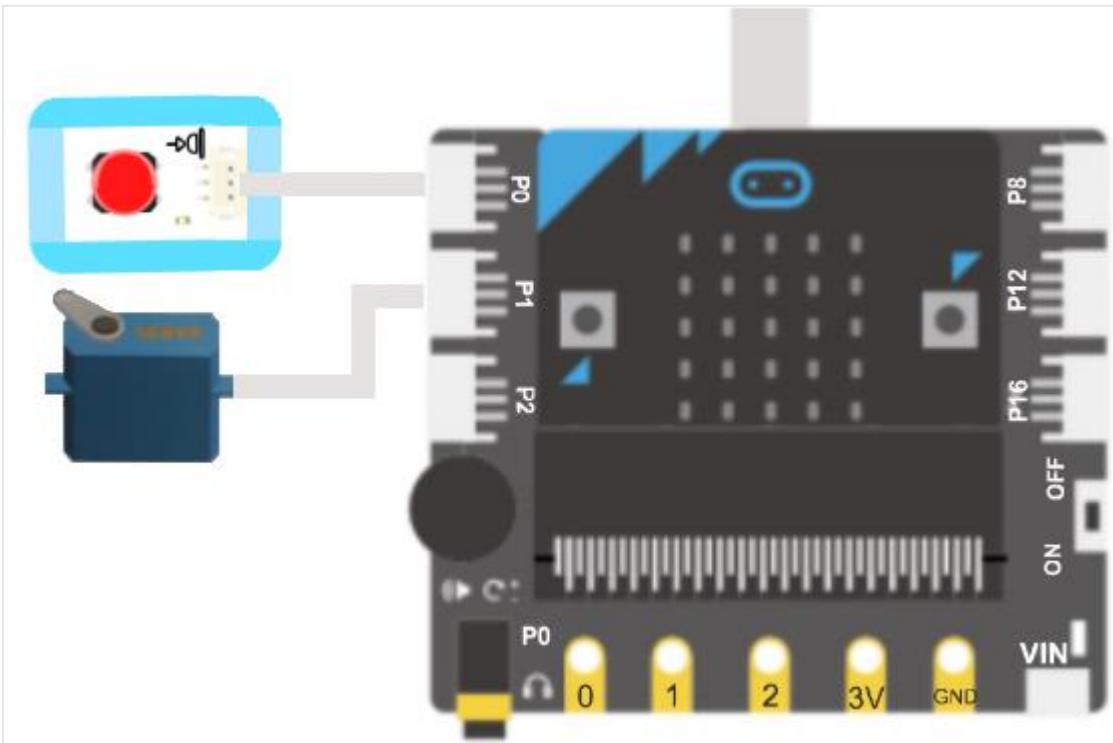
Potrebni moduli

Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 2 kabela za spajanje modula, tipkalo, servo motor

Spajanje modula

Oba modula, servo motor i tipkalo, povežite na bilo koji izvod. Pripazite samo da i u programske kôdu označite izvode kao u spoju. U ovom primjeru tipkalo je spojeno na izvod P0, a servo motor na izvod P1.

Prikaz spajanja



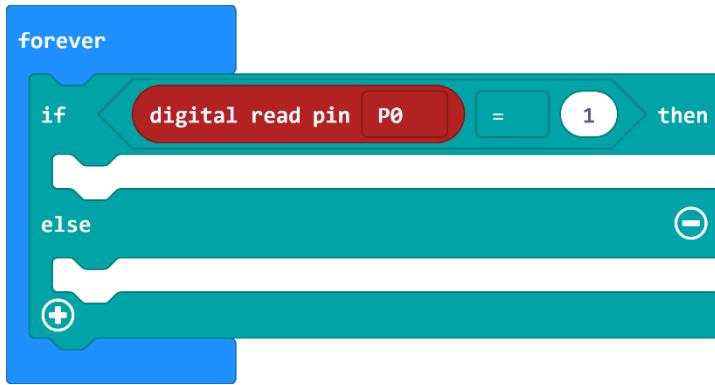
Izrada programa

Potrebne naredbe: *forever, digital write pin P0 to 0, if then else, show icon, servo write pin P0 to 180, plot xy, unplot xy, pick random 0 to 10, pause (ms)*

U *forever* petlji smjestite *if then else* naredbu u kojoj ćete definirati u kojem položaju mora biti polugica servo motora ovisno o stanju izvoda na koji je spojeno tipkalo.



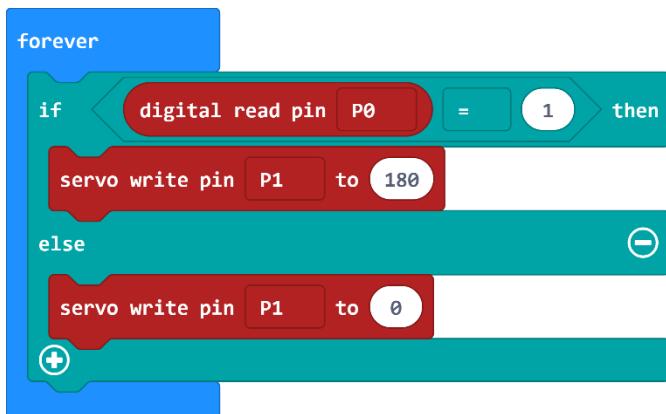
Ako je tipkalo pritisnuto, stanje njegova signala postaje 1. Iz kategorije **Logic** odaberite *naredbu usporedbe (jednako)* i smjestite ju u *if* dio naredbe. S lijeve strane naredbe smjestite naredbu *digital read pin P0*, a s lijeve upišite broj 1. Time je u programskom kôdu definiran uvjet koji provjerava je li pritisnuto tipkalo.



Ako je taj uvjet istinit, servo motor koji je spojen na izvodu P1, neka se okrene udesno za 180 stupnjeva. Naredbu za pokretanje servo motora pronađite u kategoriji **Pins**. Naredba se zove *servo write pin P0 to 180*.



Smjestite naredbu u *then* dio naredbe i promijenite izvod P0 u P1 (jer je na izvodu P1 spojen servo motor) te ostavite vrijednost argumenta 180. U *else* dio naredbe također postavite naredbu *servo write pin P0 to 180*, postavite P1, a argument promijenite iz 180 u 0. Time ste u programu odredili ako je tipkalo pritisnuto (stanje digitalnog signala je 1), polugica servo motora se pomiče udesno, a ako tipkalo nije pritisnuto tada se servo motor vraća u početnu poziciju i u njoj ostaje do ponovnog pritiska na gumb.

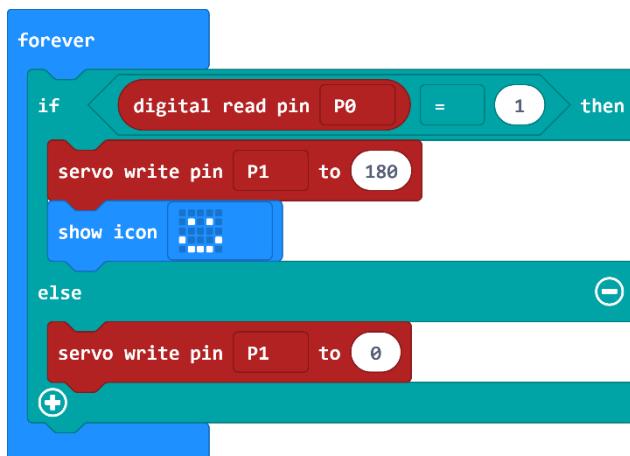


Prebacite sad program na micro:bit i provjerite kako se zaokreće polugica servo motora. Ako se previše zaokrene, smanjite vrijednost 180 na manju (primjerice 150) da polugica dođe u željenu poziciju.

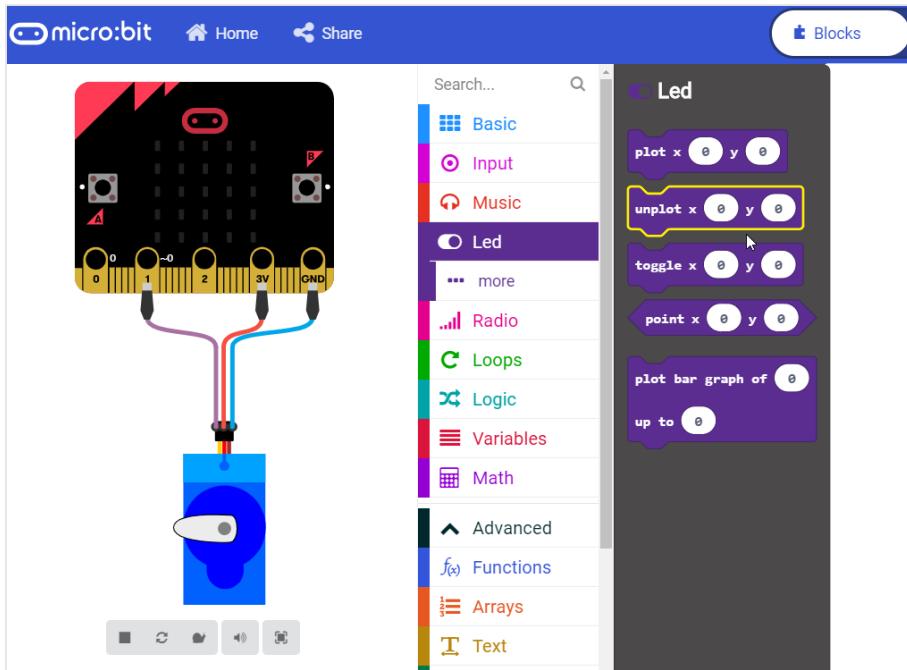
Dodajte naredbu *show icon* iz kategorije **Basic** u then dio.



Pritiskom na strelicu uz ikonu srca otvorit će se prozor s ostalim ikonama. U njemu odaberite ikonu smajlića.



U *else* dio treba dodati naredbe za nasumično paljenje i gašenje pojedinačnih svjetlećih dioda na zaslonu micro:bita. U kategoriji **Leds** nalaze se naredbe *plot xy* (paljenje dioda) i *unplot xy* (gašenje dioda).



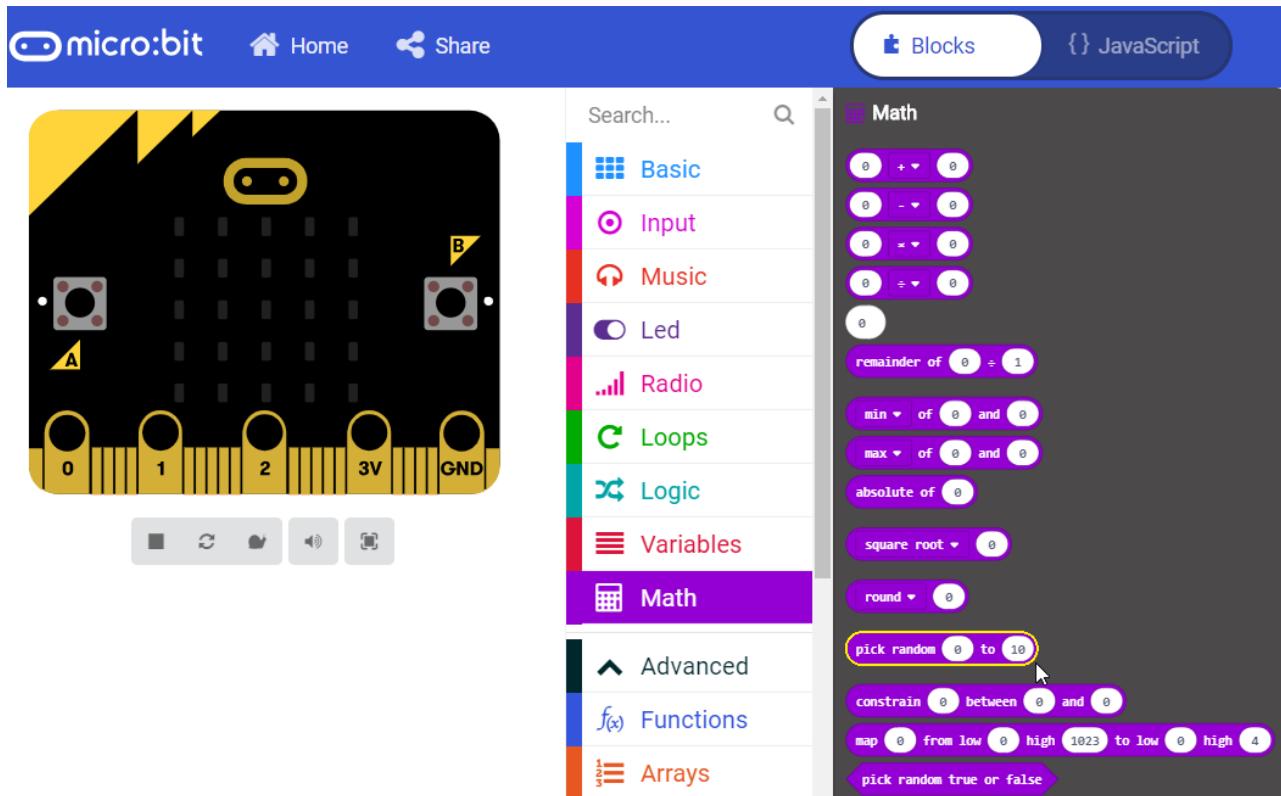
Obj je naredbe smjestite u **else** dio te iza svake od njih dodajte pauzu od 100 milisekundi. Naredba za pauzu nalazi se u kategoriji **Basic**.

```

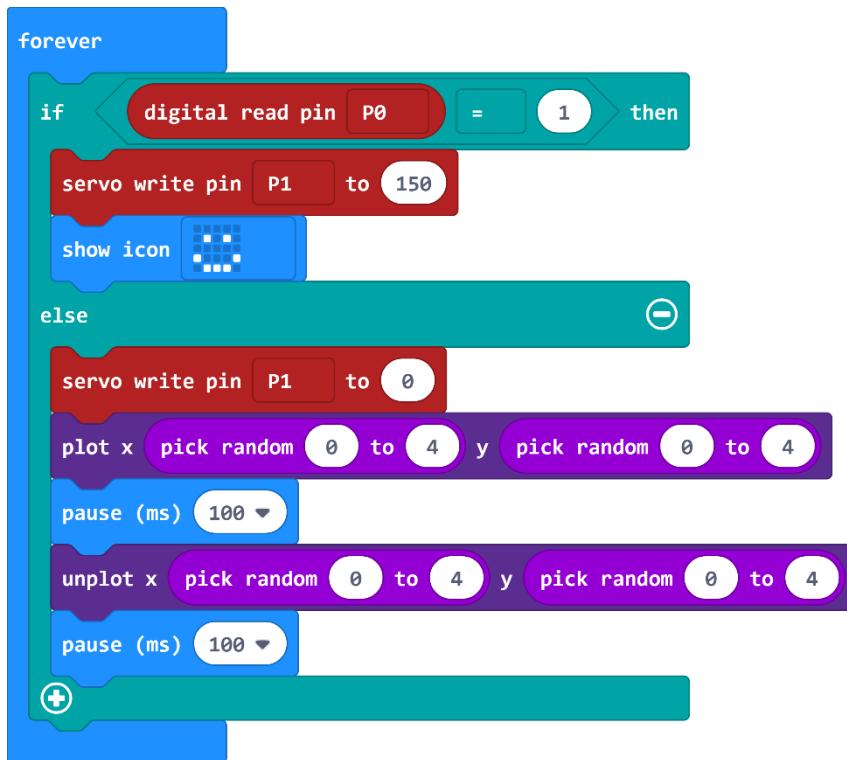
forever
  if digital read pin P0 = 1 then
    servo write pin P1 to 180
    show icon [grid]
  else
    servo write pin P1 to 0
    plot x 0 y 0
    pause (ms) 100
    unplot x 0 y 0
    pause (ms) 100
  end
end
  
```

Argument 0 uz x i y predstavlja poziciju svjetleće diode na zaslonu micro:bita. Vrijednost uz parametar x odnosi se na broj stupca u kojem se dioda nalazi, a vrijednost uz parametar y na broj retka. Ukupno je 5 stupaca i 5 redaka, no prvi stupac i redak se broje kao 0, drugi kao 1 i tako do posljednjih redaka i stupaca koji se broje kao 4. Kad miš postavite iznad određene diode na simulatoru micro:bita prikazat će se vrijednosti njezine pozicije po x i y osi.

Trenutno se pali i gasi samo prva dioda jer je njezina pozicija 0, 0. Da bi se palile i gasile ostale diode i to slučajnim redoslijedom iz kategorije **Math** odaberite naredbu *pick random 0 to 10*.



Odaberite tu naredbu i postavite je na pozicije uz x i y u *plot* i *unplot* naredbi. Umjesto 10 upišite 4 jer micro:bit ima 5 stupaca i 5 redaka. Time ste dovršili program za ovaj projekt.



Kreativni kutak

Osmislite svoje lijepe poruke ili upotrijebite neku od naših.

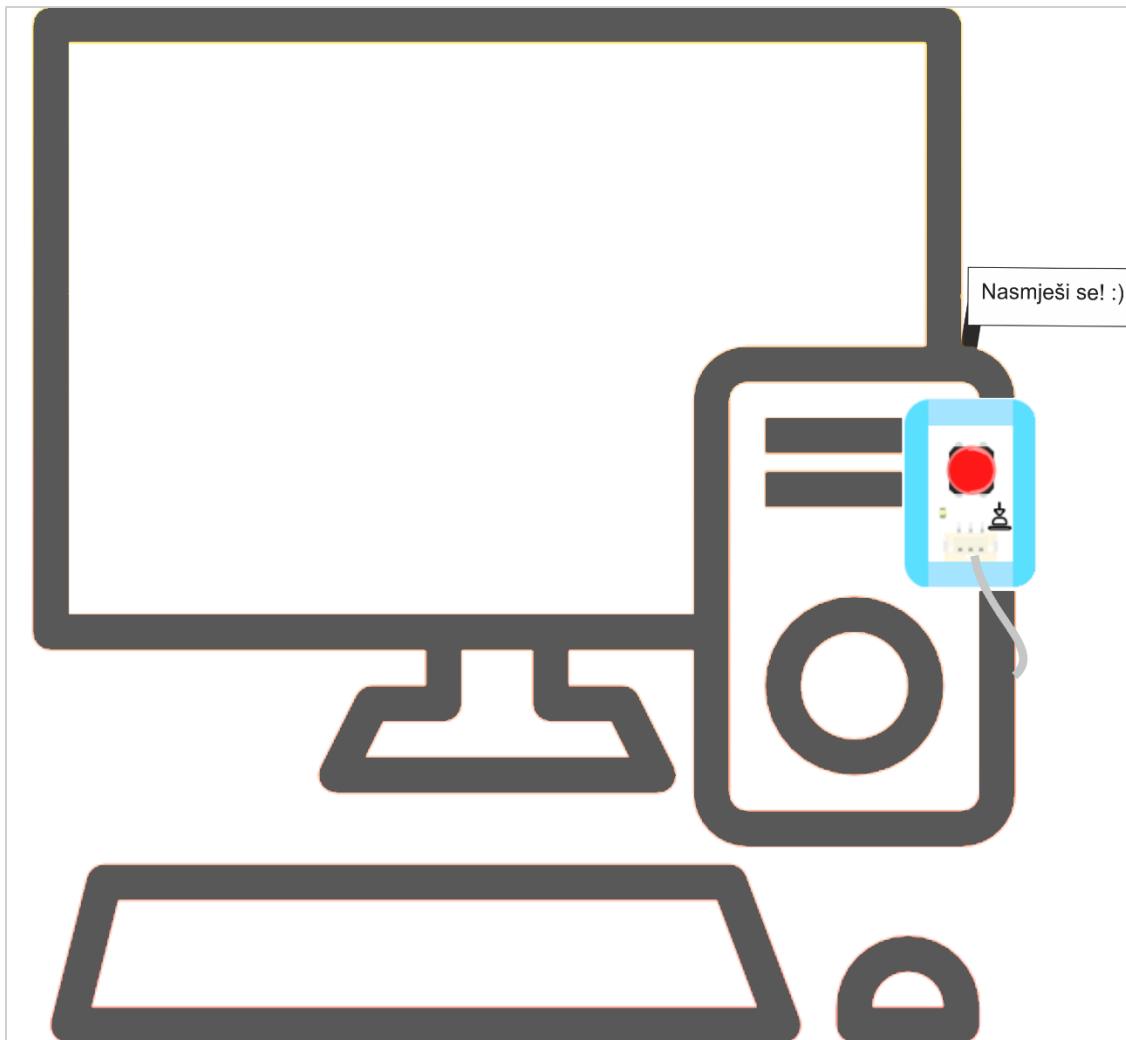
Nasmiješi se! ☺

Ispričaj nekome vic!

Poslušaj omiljenu pjesmu.

Napišite poruku i zaljepite selotejpom ju na polugicu.

Smjestite spoj blizu mesta gdje radite ili učite. Mi smo ga smjestili iza monitora. Svaki put kad pritisnemo gumb, izviri poruka pokraj monitora i izmami nam osmjeh na lice.



Projekt 6: Uređaj za praćenje učenja i pauze

Opis projekta

Kad planirate kako uskladiti svoje obaveze učenja i pisanja zadaće, pripremanja važnog referata ili izrade plakata, važno je dobro organizirati vrijeme i ne zaboraviti se odmoriti između aktivnosti kako biste sve uspješno odradili. Pomoću Boson seta izradite svjetleći sustav kojim ćete pratiti koliko dugo učite i kad je vrijeme za pauzu. Upotrijebite tipkalo i RGB LED traku za izradu ovog projekta. Pritisom na tipke A i B odredit ćete vrijeme trajanja učenja i vrijeme trajanja pauze. Učenje neka traje najmanje 20 minuta, a najviše 45 minuta. Pauza neka traje najmanje 5, a najviše 15 minuta. Protresanjem micro:bita provjerite vrijeme trajanja učenja i pauze jer će se ono ispisati na ekranu. Pritisom na tipkalo u spoju kreće mjerjenje vremena. Novi modul u ovom projektu je RGB LED traka. Ona je izlazni modul i sastoji se od 7 svjetlećih dioda koje mogu svijetliti različitim bojama RGB spektra. Izradom programa za ovaj projekt upoznat ćete neke od naredbi za programiranje trake. Za to će biti potrebno dodati poseban paket naredbi Neopixel. U programu neka traka svijetli žuto dok ne krene mjerjenje vremena. Kad krene učenje traka neka promjeni boju iz žute u zelenu. Kad završi učenje i krene pauza, traka neka svijetli plavom bojom. Na zaslonu micro:bita neka se za vrijeme učenja prikazuje slovo L, a za vrijeme pauze slovo P.

U kreativnom kutku pogledajte ideju kako izraditi i ukrasiti kutijicu u kojoj su smješteni svi elementi spoja i koja s njima predstavlja uređaj za praćenje učenja i pauze.

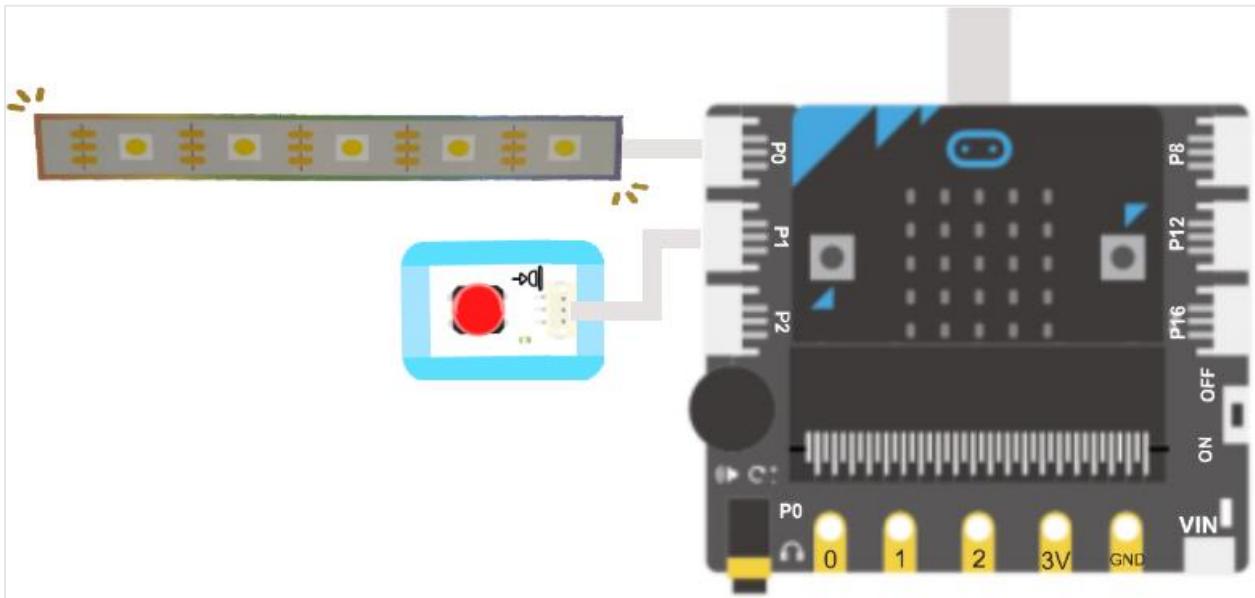
Potrebni moduli

Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 2 kabela za spajanje modula, tipkalo, RGB LED traka

Spajanje modula

U pločicu smjestite micro:bit. Na mjestima za spajanje elektroničkih modula kablovima povežite i spojite tipkalo te RGB LED traku kao što vidite na prikazu spoja. Tipkalo je digitalni ulazni modul pa se može spojiti na bilo koji izvod na pločici, a u ovom primjeru spojite ga na izvod P1. RGB LED traku spojite na izvod P0. USB kabelom povežite micro:bit s računalom kako bi spoj dobio napajanje i kako biste prebacili program na micro:bit.

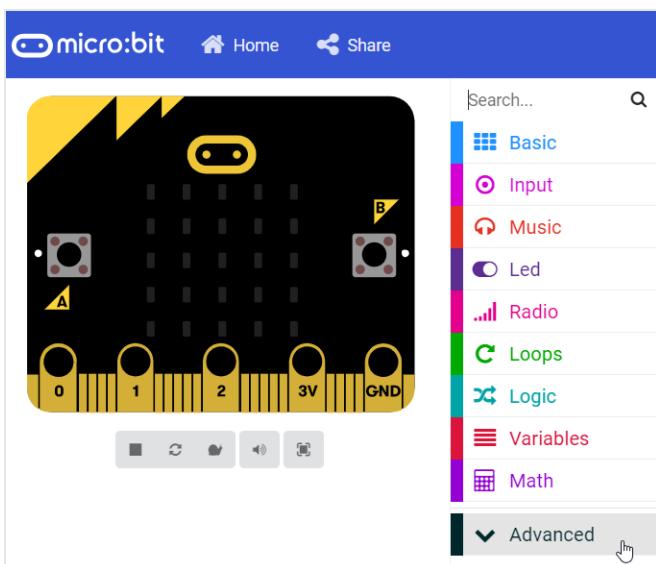
Prikaz spajanja



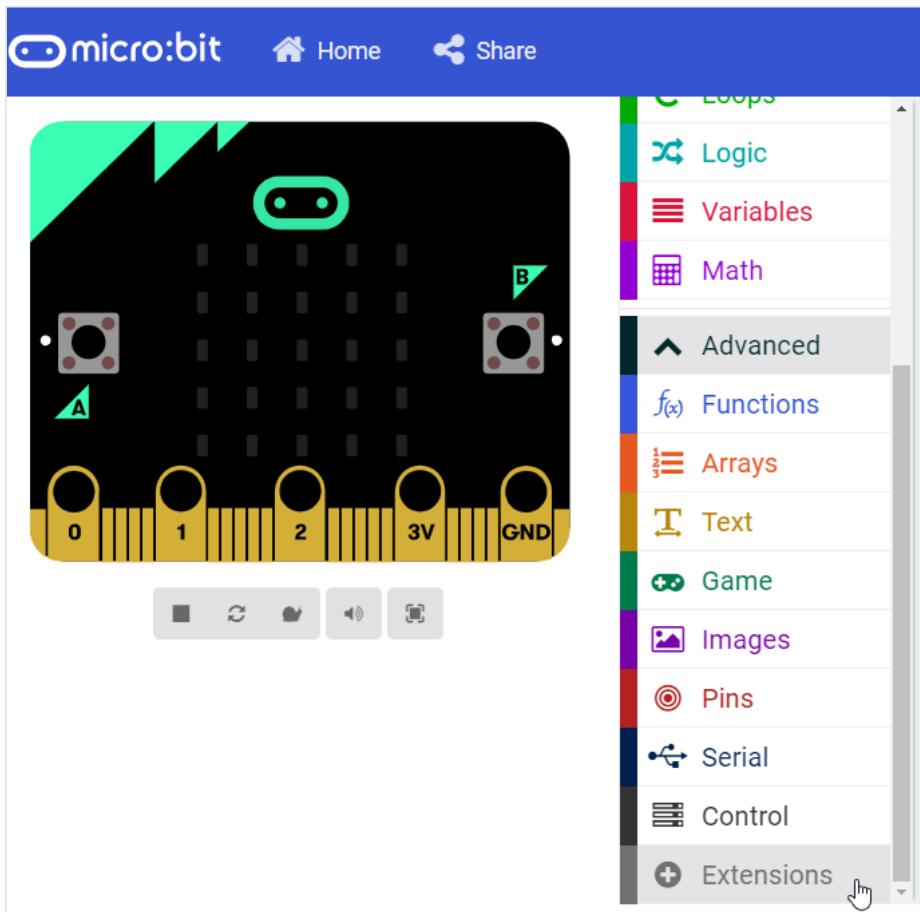
Izrada programa

Potrebne naredbe: *on start, forever, set variable to 0 change variable by, on button A pressed, on button B pressed, on shake, if then else, show number, pause (ms), digital read pin, show leds, set strip to, strip show color, clear screen*

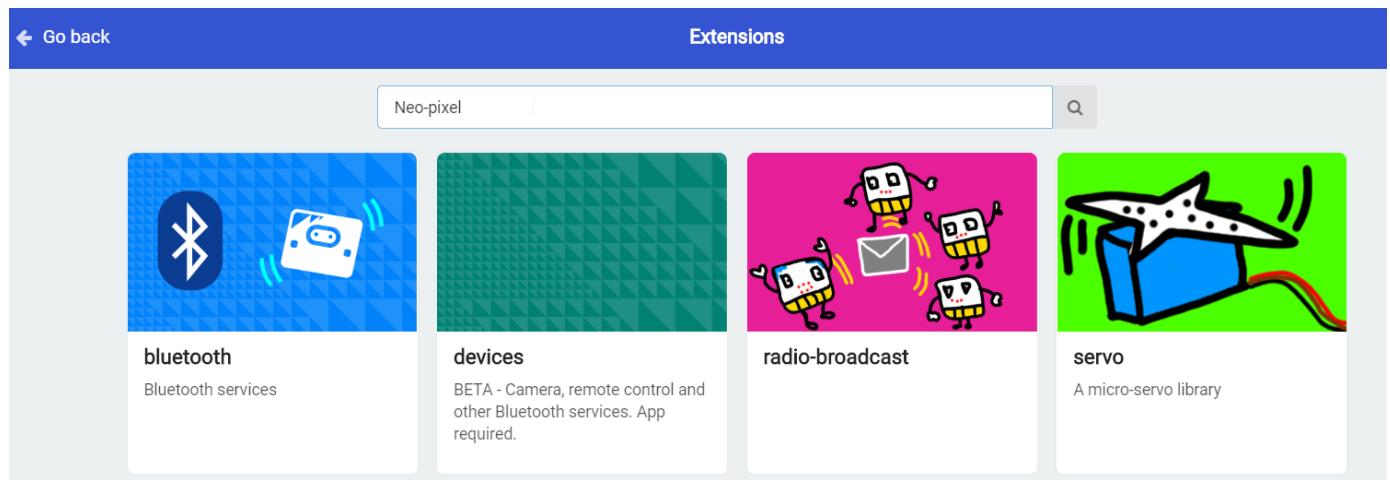
Za početak izrade ovog programa potrebno je dodati kategoriju Neopixel koja sadrži naredbe potrebne za programiranje RGB LED trake. U dijelu s kategorijama naredbi odaberite **Advanced**.



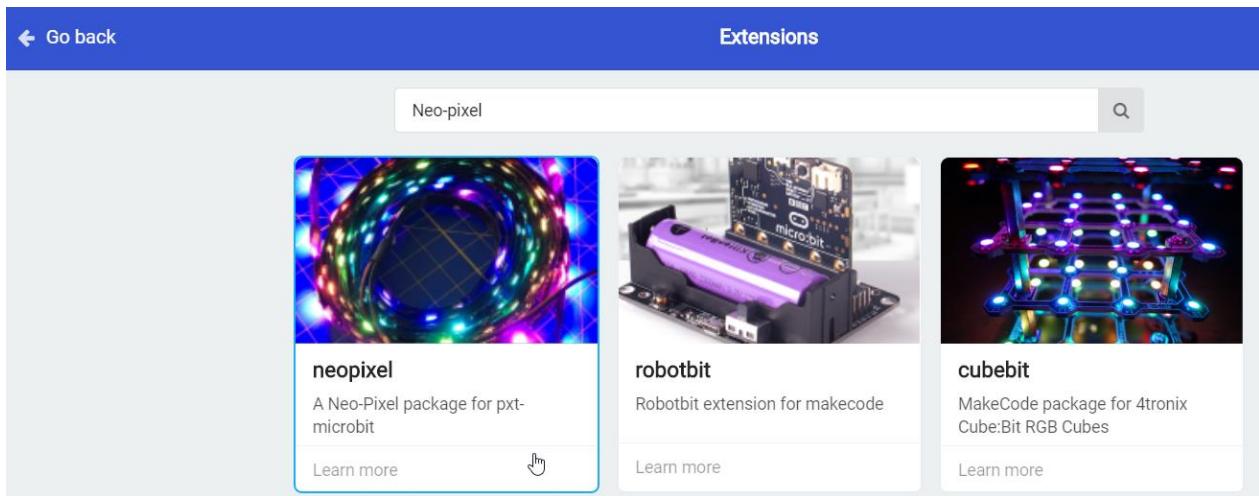
Pritiskom na **Advanced** otvara se popis s dodatnim kategorijama. Na kraju tog popisa nalazi se opcija **Extensions**. Pritisnite na nju.



Pritiskom na **Extensions** otvara se prozor preko kojeg možete pronaći potrebnu ekstenziju. U tražilicu upišite *Neo-pixel* i pritisnite enter.



U prozoru će se nakon toga pokazati popis ekstenzija koje odgovaraju upisanoj riječi u tražilici. Odaberite prvi ekstenziju kao što je prikazano na slici.



Tim odabirom ćete se vratiti u editor za programiranje te ćete u popisu kategorija pronaći dodanu kategoriju **Neopixel**. U toj kategoriji nalaze se naredbe za programiranje RGB LED trake.

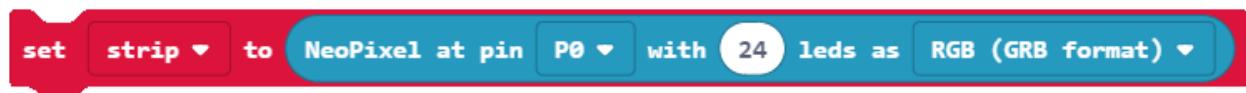
A screenshot of the MakeCode editor interface. On the left, there is a preview of a micro:bit board with various pins labeled (A, B, 0, 1, 2, 3V, GND). Below the preview are several small icons. In the center, there is a sidebar with a search bar and a list of categories: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, Neopixel (which is highlighted with a blue background), and Advanced. On the right, the main workspace shows a list of blocks under the "Neopixel" category. The blocks include: "set strip to NeoPixel at pin P0 with 24", "set range to strip range from 0 with", "strip show color red", "strip show rainbow from 1 to 360", "strip show bar graph of 0 up to 255", "strip show", "strip clear", and "hue 0 saturation 0 luminosity 0".

Sada imate sve potrebno za početak rada.

Dijelove programa, poput varijabli koje se odnose na praćenje vremena učenja i pauze, smjestit ćete u *on start* naredbu (iz kategorije **Basic**) koja se već nalazi u prostoru za programiranje.



Na početku je potrebno definirati da u programu koristite RGB LED traku. To ćete učiniti odabirom prve naredbe (*set strip to*) iz kategorije **Neopixel**.



Naredbu *set strip to* smjestite u *on start* naredbu. U dijelu naredbe gdje možete odrediti parametre, ostavite P0 jer je natom izvodu spojena traka, umjesto 24 upišite broj 7 jer se od toliko dioda traka sastoji i kao format ostavite zadanu vrijednost RGB (GRB format).



Nadalje, u ovom dijelu potrebno je definirati 3 varijable kojima se regulira praćenje i trajanje vremena za učenje i pauzu. Detaljnije upute kako kreirati varijablu pogledajte u projektu Ventilator.

Prvu varijablu nazovite *start* i dodijelite joj vrijednost 0. Na početku programa prije nego se pritisne tipkalo vrijeme se ne mjeri, stoga varijabla start ima vrijednost 0.

Druga varijabla neka se zove *ucenje* i na početku joj postavite vrijednost 20 jer je to najkraće vrijeme (u minutama) koliko učenje treba trajati.

Treća varijabla je *pauza* i njezina vrijednost na početku programa neka bude 5.

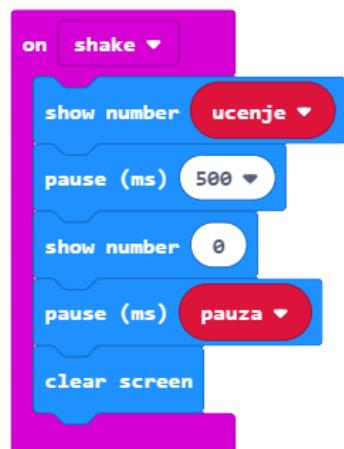
Sve varijable s njihovim pravilno definiranim vrijednostima postavite u *on start* blok.



Iz kategorije **Input** odaberite *on shake* naredbu.



U nju smjestite naredbe za prikaz vrijednosti varijabli *ucenje* i *pauza*. Za to vam je potrebna *show number* iz kategorije **Basic**, te na mjesto argumenta naredbe smjestite vrijednosti varijable *ucenje* i varijable *pauza* iz kategorije **Variables**. Neka razmak između prikaza te dvije vrijednosti bude pola sekunde (naredba *pause (ms)*) i nakon toga neka se izbriše prikaz na zaslonu micro:bita (naredba *clear screen* u dijelu *more* iz kategorije **Basic**).



Svaki put kad protresete micro:bit vidjet ćete koliko će trajati vrijeme učenja i pauze.

Vrijeme učenja i pauze možete mijenjati pritiskom na tipke A i B na micro:bitu. Neka se pritiskom na tipku A povećava vrijeme učenja za 5 dok ne dođe do 45.

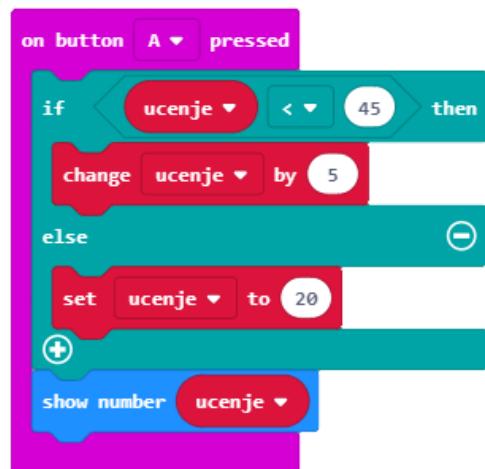
Iz kategorije **Input** odaberite naredbu *on button A pressed*.



U tom bloku definirajte uvjet da se pritiskom na tipku A povećava vrijednost varijable ucenja za 5 sve dok je ta vrijednost manja od 45, inače neka varijabla ucenje poprimi vrijednost 20. Iz kategorije **Logic** odaberite *if then else* naredbu i smjestite ju u blok *on button A pressed*.

Za mijenjanje vrijednosti varijable upotrijebite naredbu *change ucenje by* iz kategorije **Variables**. Za postavljanje vrijednosti varijable *ucenje* na 20, koristite naredbu *set ucenje to 20* koju možete kopirati iz bloka *on start*. Umjesto argumenta 1, upišite 5.

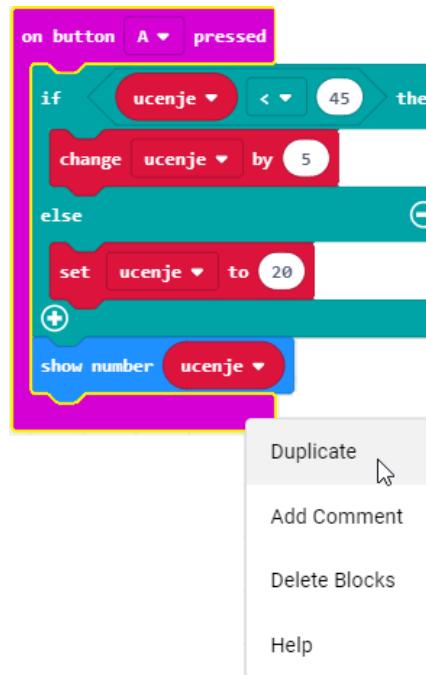
Na kraju u bloku *on button A pressed* dodajte naredbu za prikaz trenutne vrijednosti varijable *ucenje*. Za to upotrijebite naredbu *show number* iz kategorije **Basic**, te na mjesto argumenta naredbe smjestite vrijednost varijable *ucenje* iz kategorije **Variables**.



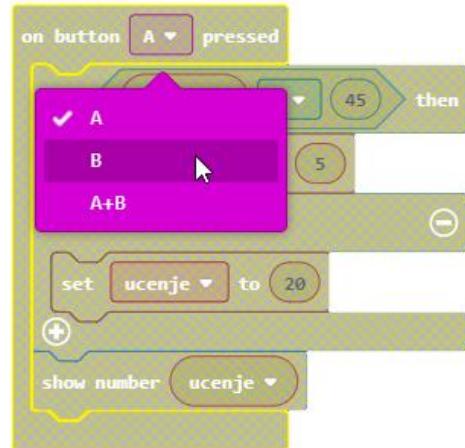
Isti postupak je potreban za mijenjanje vrijednosti varijable pauza.

Da biste si skratili vrijeme, kopirajte blok on button A pressed jer on sadrži sve potrebne naredbe, samo je potrebno napraviti nekoliko izmjena.

Kopirajte blok tako da ga najprije odaberete mišem u ljubičastom području. Zatim desnim klikom miša otvorit će se prozor s opcijama. Odaberite Duplicate.



Duplicirana naredba je siva jer u programu ne mogu postojati dva on button A pressed bloka. Pritisnite na strelicu uz slovo A i odaberite B.

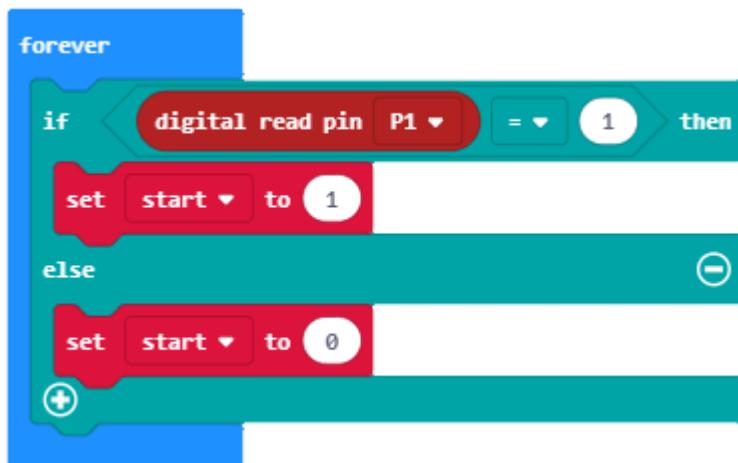


Nastavite s izmjenama u bloku za tipku B.

Pritisom na strelice uz varijablu ucenje, odaberite varijablu pauza u svakoj naredbi. U if dijelu umjesto 45 upišite 15, a u else dijelu umjesto 20 upišite 5.



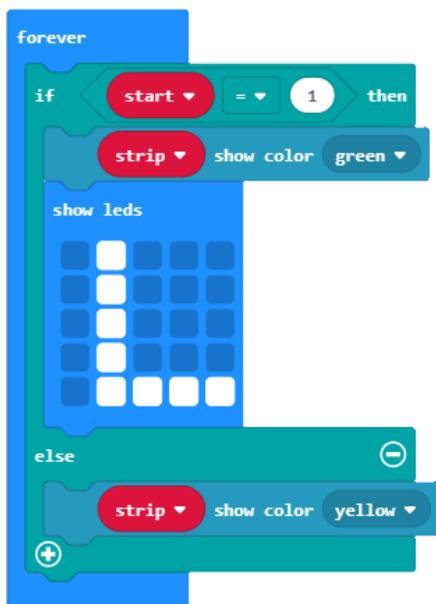
Slijedi dio u kojem programirate provjeru, je li se dogodio pritisak tipkala u spoju. Ako je tipkalo, koje je spojeno na izvod P1 bilo pritisnuto neka varijabla *start* poprimi vrijednost 1, inače neka bude 0. Provjeru tog uvjeta smjestite u *forever* petlju. Naredbu za provjeru stanja tipkala potražite u kategoriji **Pins**. To je *digital read pin*. Smjestite ju u *if* dio logičke naredbe *if then else* i promijenite P0 u P1 te operatorom provjerite njezinu vrijednost.



Prisjetite se da prema zadatku traka mora svijetliti žuto dok ne krene vrijeme (prije pritiska na tipkalo). Za vrijeme učenja ona svijetli zeleno, a za vrijeme pauze plavo. Za paljenje svjetla na traci upotrijebite naredbu *strip show color* iz kategorije **Neopixel**.



U novoj forever petlji *if then else* naredbom provjerite stanje varijable *start*. Ako je vrijednost varijable *start* jednaka 1, znači da je netko pritisnuo tipkalo i može krenuti vrijeme učenja. Neka se pri tome upale zelena svjetla na RGB LED traci, a na zaslonu micro:bita neka se prikazuje slovo L. Za prikaz slova L možete upotrijebiti *show leds* naredbu iz kategorije **Basic** i iscrtati slovo. Inače neka traka svijetli žuto kao na početku.

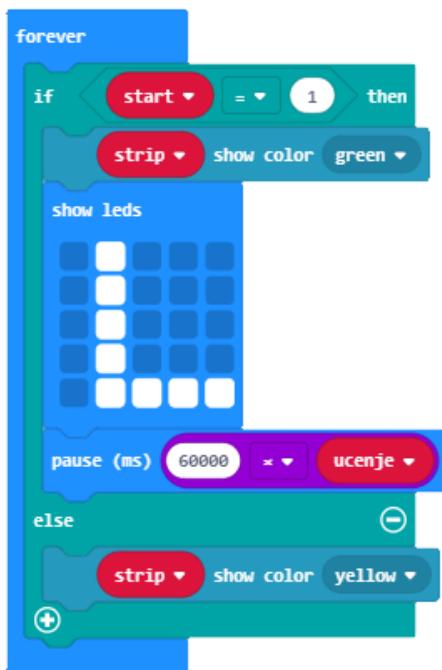


Nakon *show leds* naredbe potrebno je postaviti naredbu koja prati vrijeme. Za to ćeće upotrijebiti naredbu *pause (ms)* iz kategorije **Basic**. S obzirom na to da učenje može trajati onoliko minuta koliko postavite pritiskom na tipku A (20, 25, 30, 35, 40 ili 45 minuta) tu je vrijednost potrebno postaviti kao argument naredbe *pause (ms)*.

Za to Vam je potrebna naredba za množenje iz kategorije **Math**.



Da bi se vrijednost varijable *ucenje* pretvorio u minute, pomnožite 60000 ms s vrijednosti varijable *ucenje*. Kao prvi faktor u naredbi za množenje upišite 60000, a kao drugi postavite varijablu *ucenje*. Tu cijelu naredbu postavite kao argument naredbe za pauzu.



Naredba za pauzu odgađa izvršenje naredbi koje slijede i koje je potrebno dodati, a to su naredbe za prikaz da je došlo vrijeme pauze.

Stoga u nastavku dodajte naredbu za paljenje plavog svjetla na RGB LED traci (*strip show color blue*), *show leds* naredbu kojom će se na zaslonu micro:bita prikazati slovo P i naredbu za pauzu čiji argument će biti minuta (60000 ms) pomnožena s vrijednosti varijable *pauza*.

U *else* dio kod naredbe da traka svijetli žuto, dodajte i naredbu *clear screen*.

Time ste dovršili program, a gotov kôd pogledajte na slici u nastavku.

on start

```

set strip to NeoPixel at pin P0 with 7 leds as RGB (GRB format)
set start to 0
set ucenje to 20
set pauza to 5

```

on shake

```

show number ucenje
pause (ms) 500
show number pauza
pause (ms) 500
clear screen

```

on button A pressed

```

if ucenje < 45 then
  change ucenje by 5
else
  set ucenje to 20
show number ucenje

```

forever

```

if digital read pin P1 = 1 then
  set start to 1
else
  set start to 0

```

on button B pressed

```

if pauza < 15 then
  change pauza by 5
else
  set pauza to 5
show number pauza

```

forever

```

if start = 1 then
  strip show color green
show leds

```

pause (ms) 60000 x ucenje

```

strip show color blue
show leds

```

pause (ms) 60000 x pauza

```

else
  strip show color yellow
  clear screen

```

Kreativni kutak

Ukrasite kutiju s motivima učenja i izradite svoj jedinstveni uređaj za praćenje učenja i pauze.

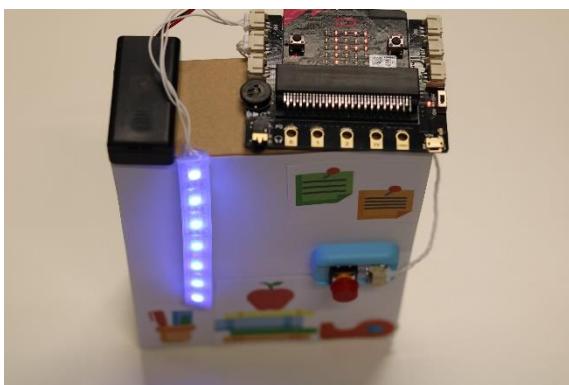
Prije učenja:



Učenje:



Pauza:



Projekt 7: Disco light

Opis projekta

Nastavite u ovom projektu rad s RGB led trakom i senzorom za zvuk. Izradite *disco light*. *Disco light* pokazuje paljenjem svjetla dioda na traci koliko je glasno u prostoru. Praćenjem glasnoće senzor za zvuk će preko micro:bita slati signal na traku koja će redom paliti svoje svjetleće diode ovisno o tome koliko je glasan zvuk u prostoru. Što je glasnije, to će se više dioda upaliti. Boje kojom će diode svijeliti odredit ćete novom naredbom *show rainbow*.

Ovaj uređaj i program možete iskoristiti na različitim zabavama za disco doživljaj.

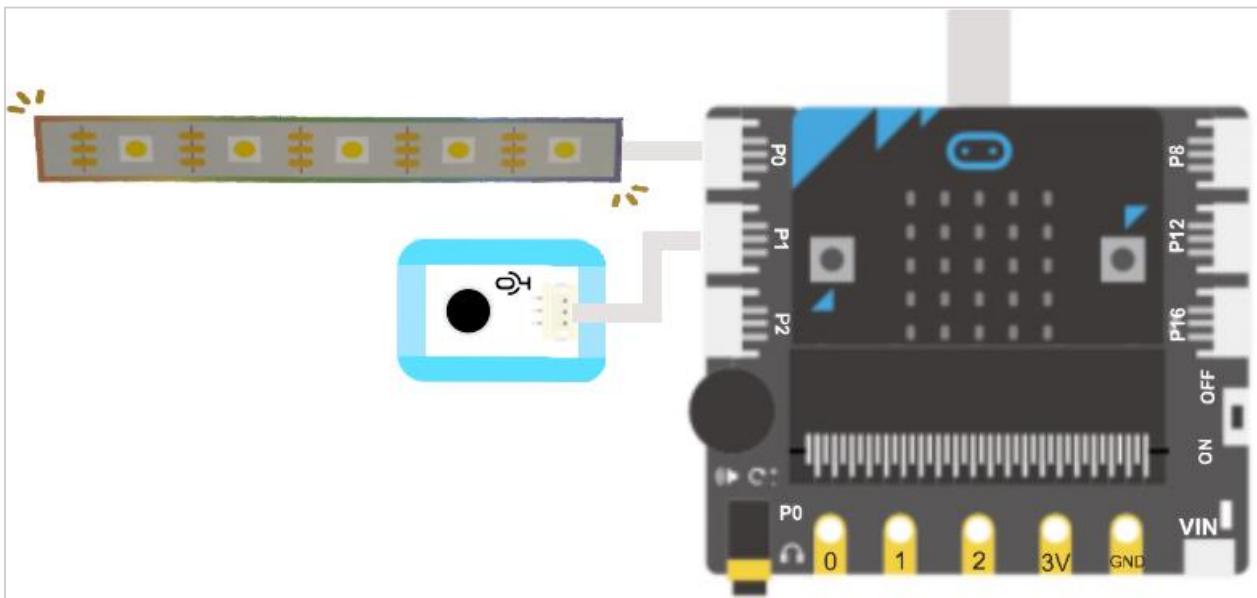
Potrebni moduli

Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 2 kabela za spajanje modula, senzor za zvuk, RGB LED traka

Spajanje modula

Prema prikazu spajanja spojite ulazni modul senzor za zvuka na izvod P1 (jer je senzor za zvuk analogni ulazni modul), a RGB traku svjetlećih dioda spojite na izvod P1.

Prikaz spajanja



Izrada programa

Potrebne naredbe: *on start, set strip to NeoPixel pin P0 with 7 leds as, forever, set variable to, map, strip clear, strip range, show rainbow, strip show, pause (ms)*

Ako niste, dodajte ekstenziju Neopixel za rad s RGB trakom. Detaljna uputa kako dodati tu ekstenziju nalazi se u projektu 6 (Uređaj za praćenje učenja i pauze.)

Na početku programa potrebno je inicijalizirati RGB traku. U *on start* blok postavite naredbu *set strip to* iz kategorije **Neopixel**. U dijelu naredbe gdje možete odrediti parametre, ostavite P0 jer je na tom izvodu spojena traka, umjesto 24 upišite broj 7 jer se od toliko dioda traka sastoji i kao format ostavite zadalu vrijednost RGB (GRB format).

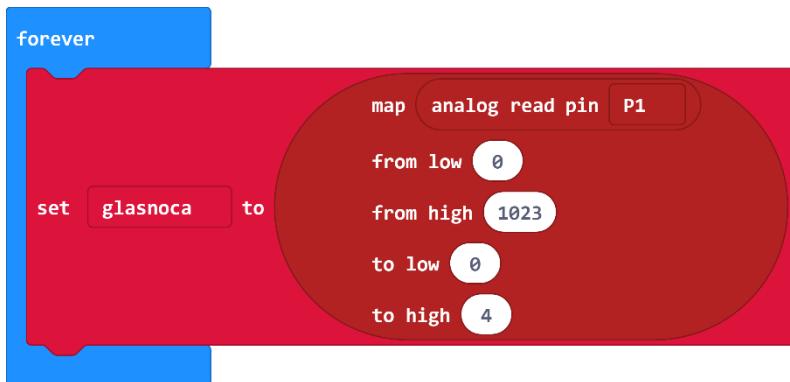


U kategoriji **Variables** izradite varijablu *glasnoca*. Njezina vrijednost neka ovisi o vrijednostima koje očitava senzor zvuka. Varijabla *glasnoca* poveznica je između vrijednosti koje očitava senzor zvuka te načina rada RGB trake svjetlećih dioda.

Traka ima ukupno 7 dioda te želite da se one pale od prve do zadnje ovisno o tome koliko je glasan zvuk u prostoru. Iz tog razloga potrebna je naredba koja će niske razine glasnoće zvuka povezati s početnim diodama na traci, a visoke razine glasnoće zvuka sa završnim diodama. To je naredba za mapiranje koju ste upoznali u projektu 2 (Reguliranje intenziteta svjetla). Pomoću te naredbe „preslikat“ će se vrijednosti senzora zvuka iz njegovog zadanog raspona u raspon svjetlećih dioda kojih ima 7.

Iz kategorije **Pins** odaberite naredbu za mapiranje i nju postavite kao vrijednost varijable *glasnoca* te to smjestite u *forever* petlju.

S obzirom da je analogna vrijednost senzora za zvuk koji je spojen na izvod P1 ona koju treba preslikati na broj dioda, u map dio naredbe postavite naredbu analog read pin P1.



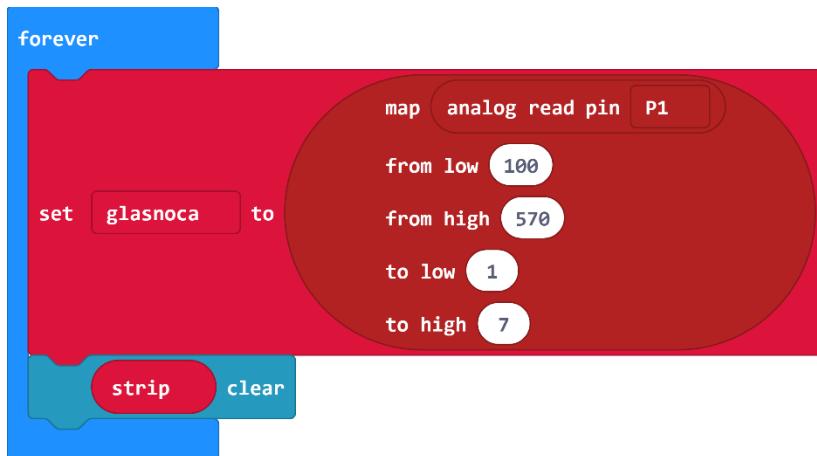
Ne želite da *disco light* reagira na sve zvukove u prostoru stoga zadajte da registrira one iznad 100, stoga u from low dio upišite 100. U from high dio upišite višu vrijednost koju će senzor zvuka registrirati kao maksimalnu glasnoću. U ovom primjeru to je 570. U dijelu to low upišite 1, a to high 7 jer je to raspon broja svjetlećih dioda na traci.



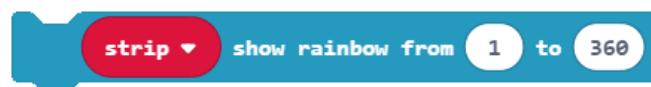
Nakon ove naredbe dodajte naredbu *strip clear* iz kategorije **Neopixel** kako bi se prije paljenja dioda prema novoj glasnoći ugasile prethodno upaljene ledice.



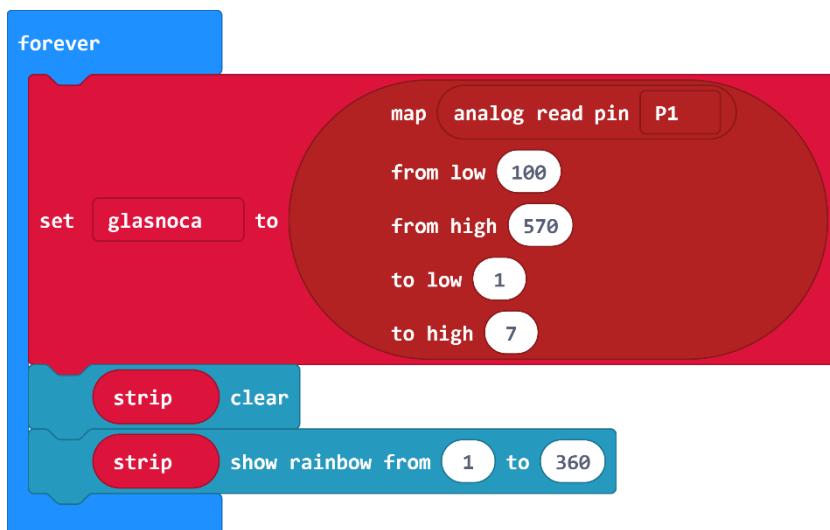
Bez te naredbe, u slučaju da je najprije senzor za zvuk detektirao glasniji zvuk pa nakon njega tiši, diode glasnijeg zvuka bi ostale upaljene i pri tišem zvuku.



No, da bi se diode uopće upalile potrebno je dodati naredbu za boju i način svjetljenja dioda. Ovog puta neka diode svijetle različitim bojama duginog spektra pomoću naredbe *show rainbow*. Ta naredba nalazi se u kategoriji **Neopixel**.



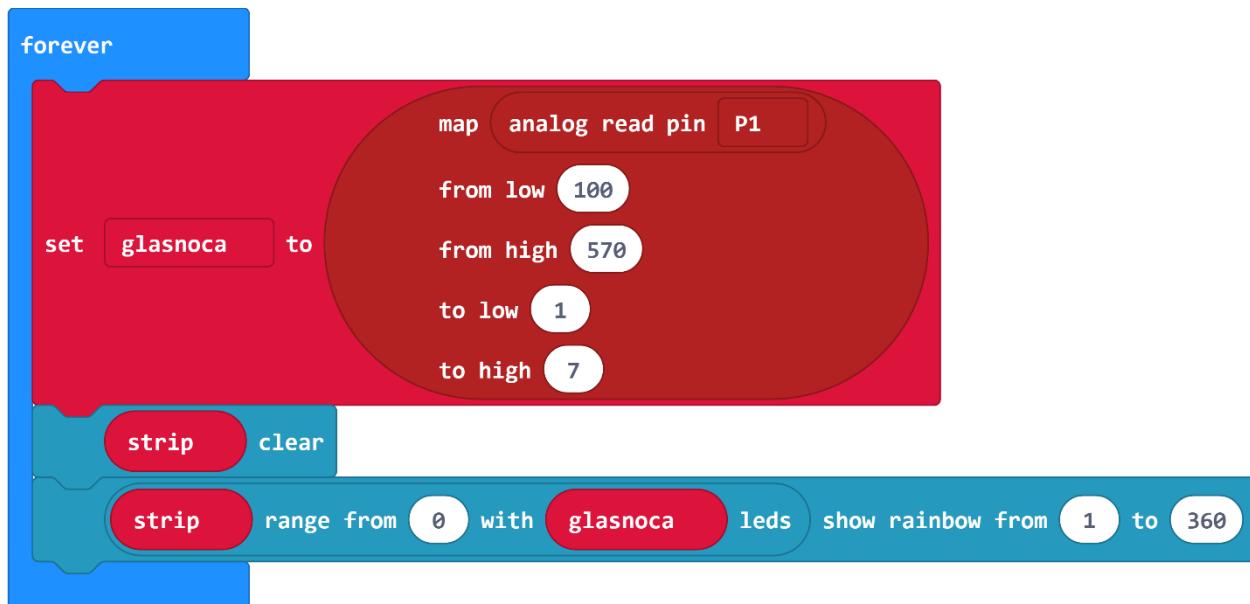
Odaberite tu naredbu i dodajete ju u forever petlji kao na slici. Brojevi od 1 do 360 predstavljaju raspon boja od crvene do plave kojima mogu svijetliti diode na RGB traci.



U naredbi *show rainbow* potrebno je promijeniti parametar *strip* kako bi se na traci palio točno određen broj dioda u ovisnosti o glasnoći zvuka. Potrebno je dodati raspon. Iz kategorije Neopixel odaberite naredbu *set range*.



Iz te naredbe uzmite dio *strip range form o with 4 leds* i postavite ga na mjesto *strip* u naredbi *show rainbow*. Taj dio naredbe određuje koliko će se stvarno dioda upaliti. To u ovom zadatku mora ovisiti o varijabli *glasnoca*, stoga nju postavite kao argument umjesto broja 4.

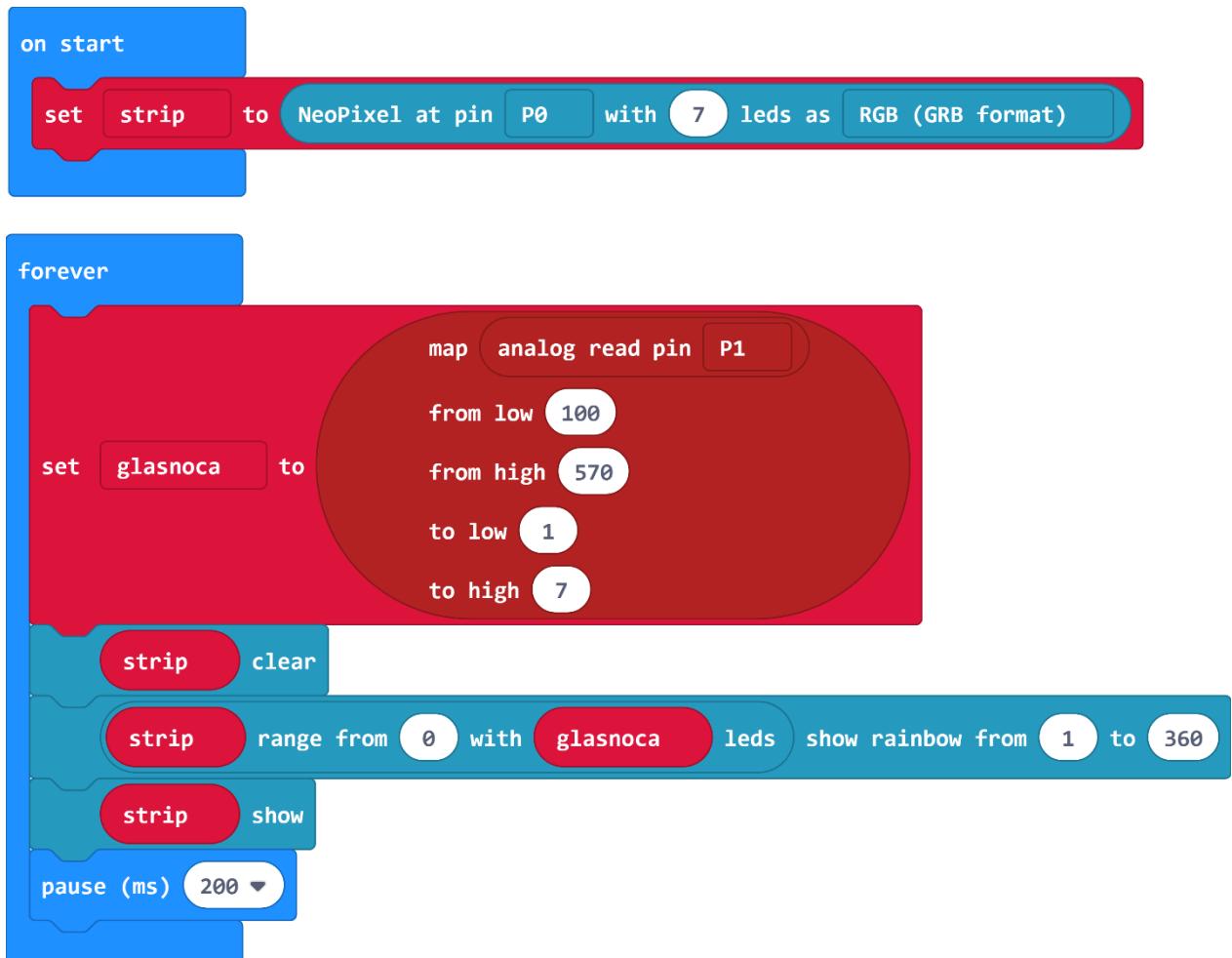


Time je definirano da se na traci pali onaj broj dioda koji koliko je glasno u prostoru i da svijetle dugim bojama.

Potrebno je još dodati naredbu za paljenje dida na traci. Za to upotrijebite *strip show* naredbu iz kategorije **Neopixel**.



Na kraju dodajte pauzu od 200 ms kako bi se mjerjenje zvuka događalo svakih 0.2 sekunde i sukladno tom vremenu palilo svjetlo na diodama trake.



Kreativni kutak



Projekt 8: Rampa na parkiralištu

Opis projekta

U novom projektu upoznat ćete rad senzora za pokret. Izradit ćete spoj u kojem će biti priključena 3 elektronička modula. Uz senzor za pokret, dodat ćete servo motor i svjetleću diodu. Ovim modulima možete izraditi model rampe na parkiralištu. Polugica na servo motoru će se pomaknuti kad senzor za pokret uoči promjenu, to jest, kad mu približi neki subjekt. Također, kad subjekt prođe, polugica će se vratiti na početni položaj. Svjetleća dioda će u tom spoju svijetliti kad je rampa podignuta,a a bit će ugašena kad je rampa spuštena. Senzor je digitalni ulazni modul.

Kreativnim idejama izradite rampu za ulazak na parkiralište. Možete spojiti rad sa flomasterima, papirima i ljepilom te lego kockama.

Potrebni moduli

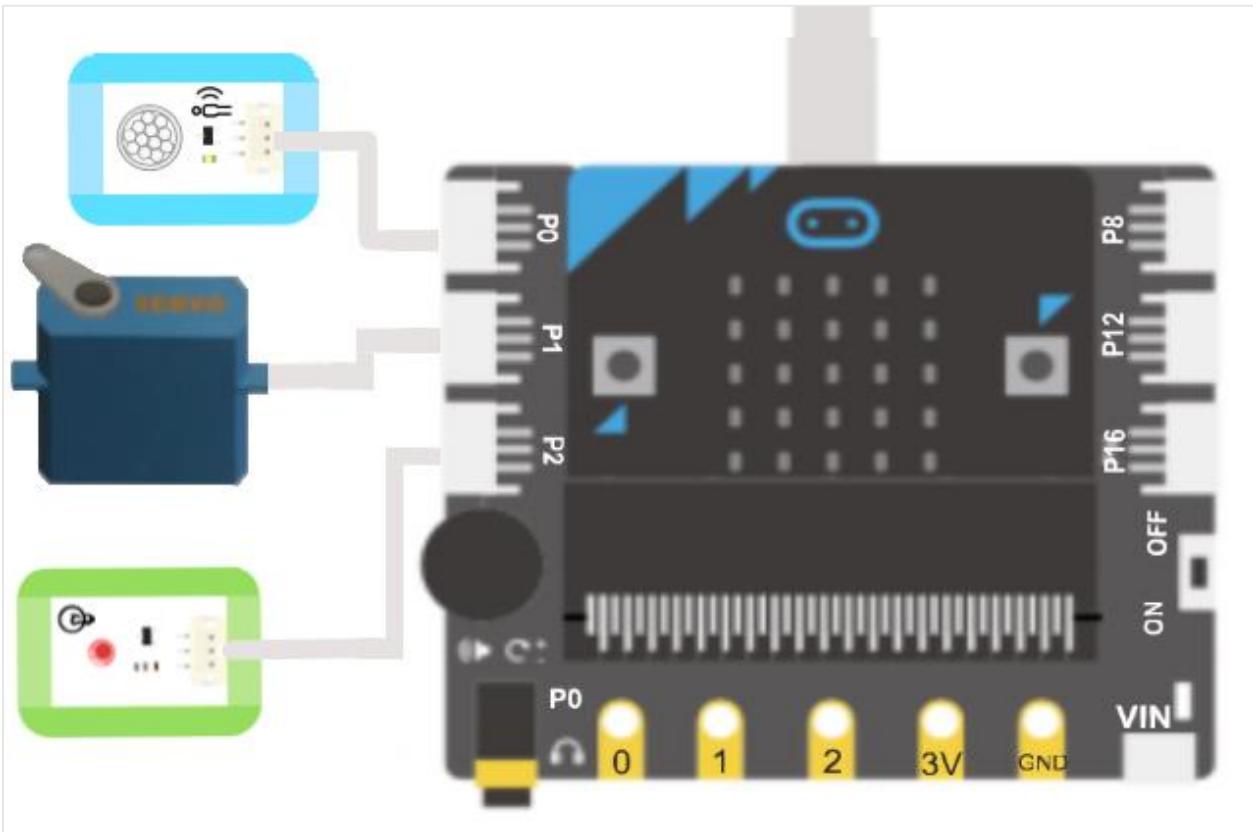
Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 3 kabela za spajanje modula, senzor pokreta, svjetleća dioda, servo motor

Spajanje modula

Prema prikazu spajanja kabelima povežite module na izvode upravljačke pločice. Možete ih rasporediti i na druga mesta ako vam tako bolje odgovara kad ćete spoj implementirati u svoj kreativni rad. Pripazite pri tome da i u programskome kôdu budu postavljeni odgovarajući izvodi jer u suprotnome se program neće izvršavati. U ovom primjeru senzor za pokret spojen je na izvod P0, svjetleća dioda na izvod P2, a servo motor na izvod P1. Na servo motor pričvrstite jednu od polugica iz seta.

Napomena: Kad završite s prebacivanjem programa na micro:bit, spojite kabel za napajanje na upravljačku pločicu jer na njoj imate priključena 3 modula.

Prikaz spajanja



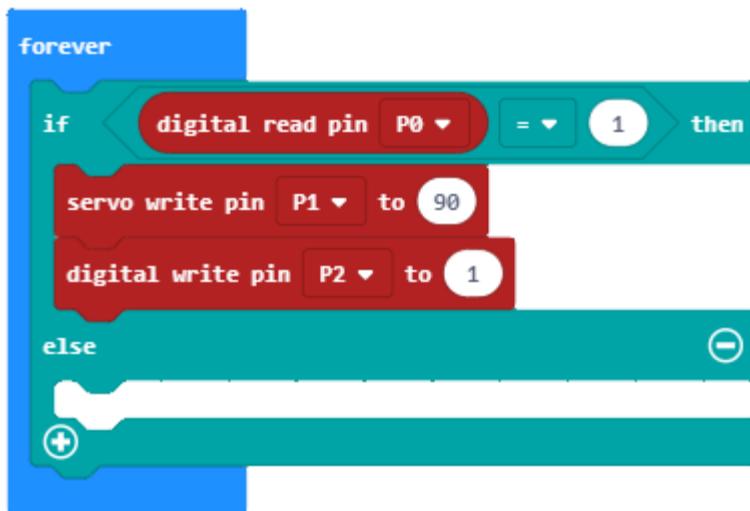
Izrada programa

Potrebne naredbe: *on start, forever, true, set variable to, if then else, digital read pin, digital write pin to, change variable by, servo write pin to, show number*

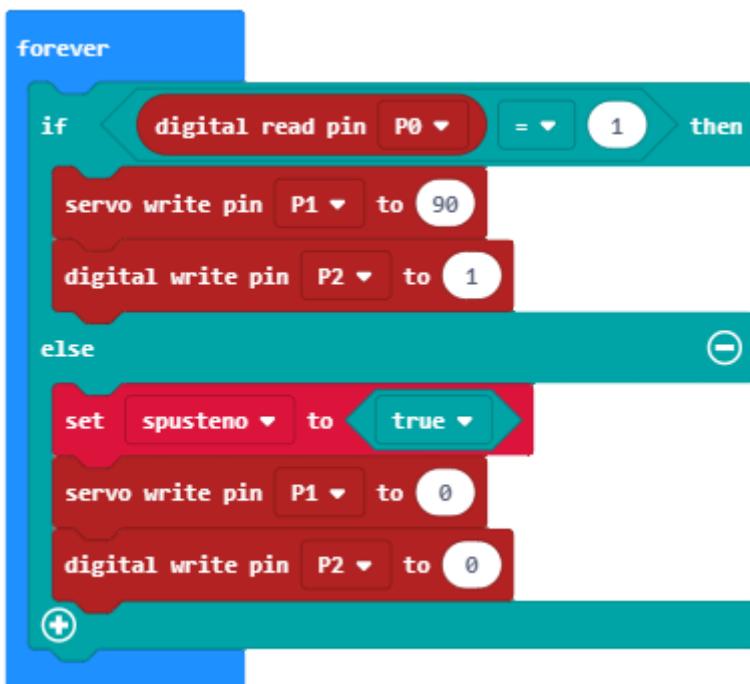
Na početku programa kreirajte varijablu *broj* u koju će se pohranjivati vrijednost koliko puta se rampa podigla. Kreirajte varijablu *spusteno* kojoj na početku dodijeljuje logičku vrijednost true iz kategorije **Logic**. Naredbe za inicijaliziranje obaju varijabli postavite u *on start* blok.



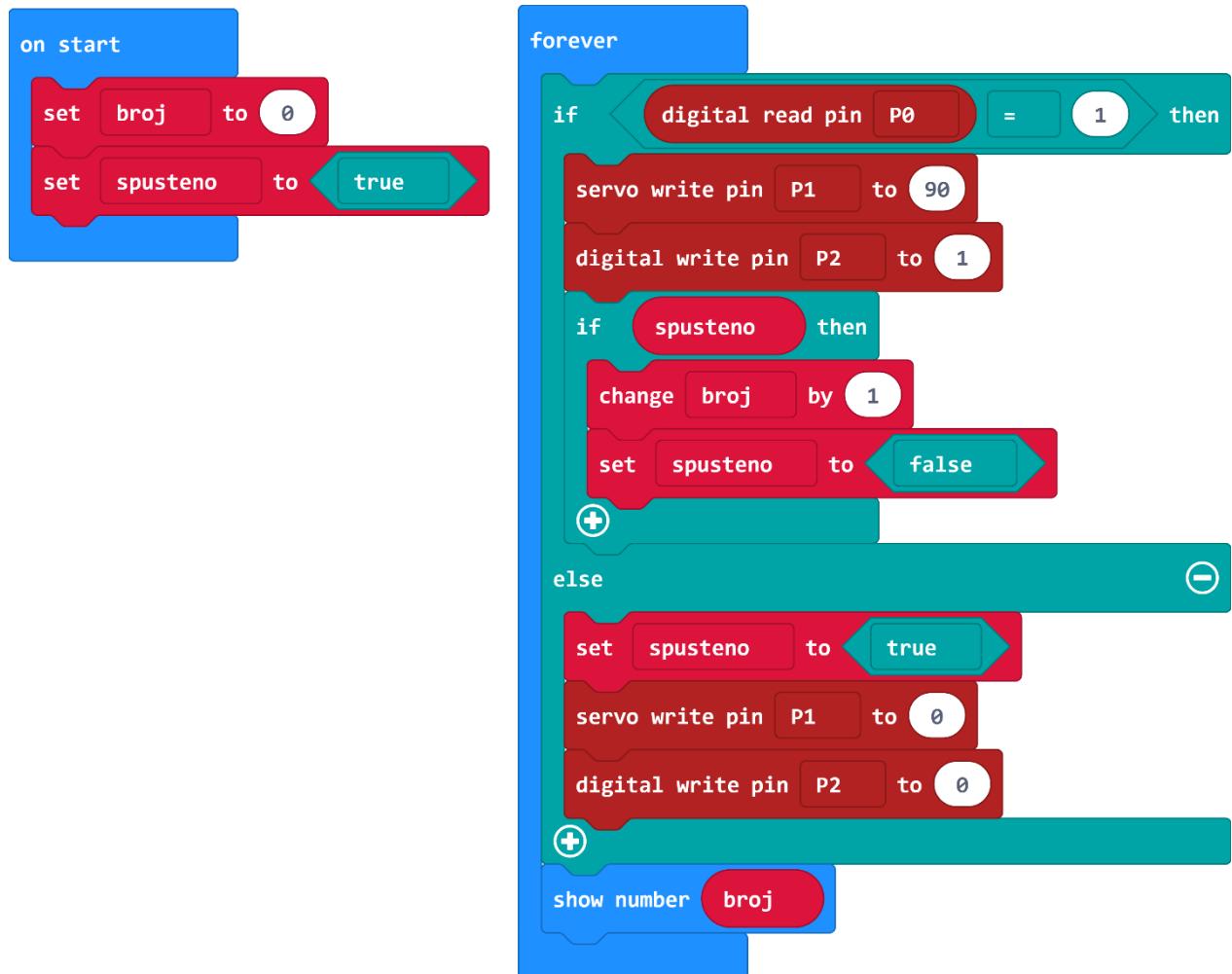
U forever petlji treba definirati uvjet kojim se ispituje je li senzor za pokret uočio primjenu. Ako se netko približio senzoru on poprima vrijednost 1, podiže se rampa (servo write pin P1 to 90) te pali svjetleća dioda (digital write pin P2 to 1).



Inače, kad senzor za pokret ne detektira promjenu, rampa je spuštena (varijabla *spusteno* je istina), servo motor je u položaju 0, a svjetleća dioda je ugašena.

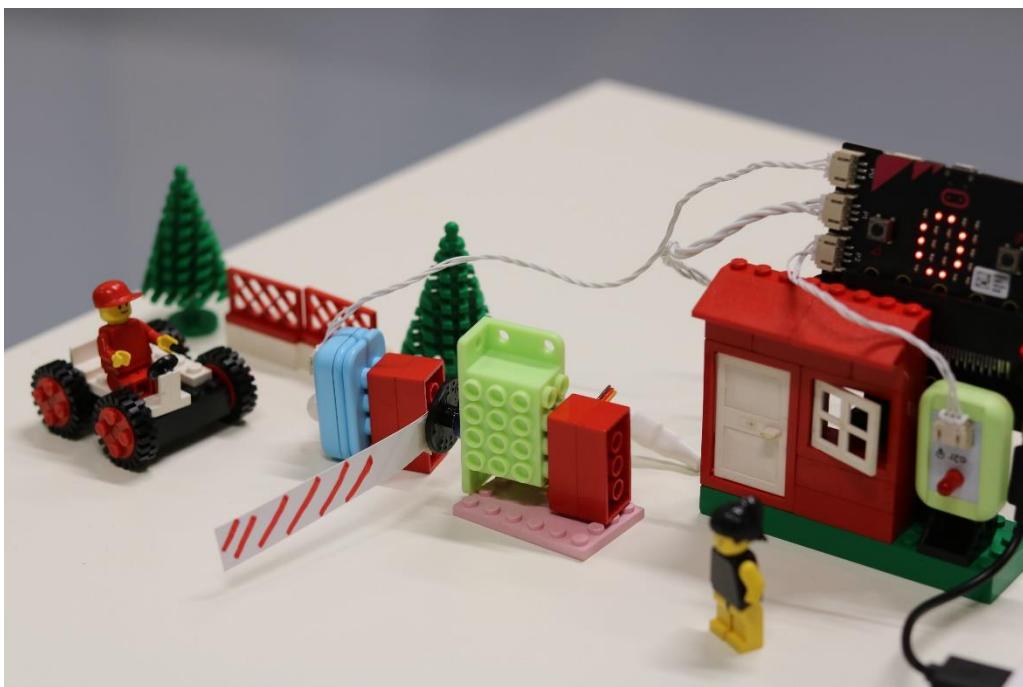


Kako bi mogli izbrojati koliko puta se podigla rampa, u *if* dijelu naredbe u *forever* petlji potrebno je ugnijezditi novu *if then* naredbu i smjestiti ju ispod postojećih naredbi. U njoj definirajte ako varijabla *spusteno* točno, tada neka se vrijednost varijable poveća za 1, a varijabla *spusteno* neka postane netočno. To je potrebno kako brojač posjetitelja ne bi broao cijelo vrijeme dok je rampa podignuta, već samo u trenutku kad se podigne (kad je varijabla *spusteno* istina), a nakon toga stanje varijable *spusteno* odmah postavljamo na neistinu da se brojanje ne nastavi.

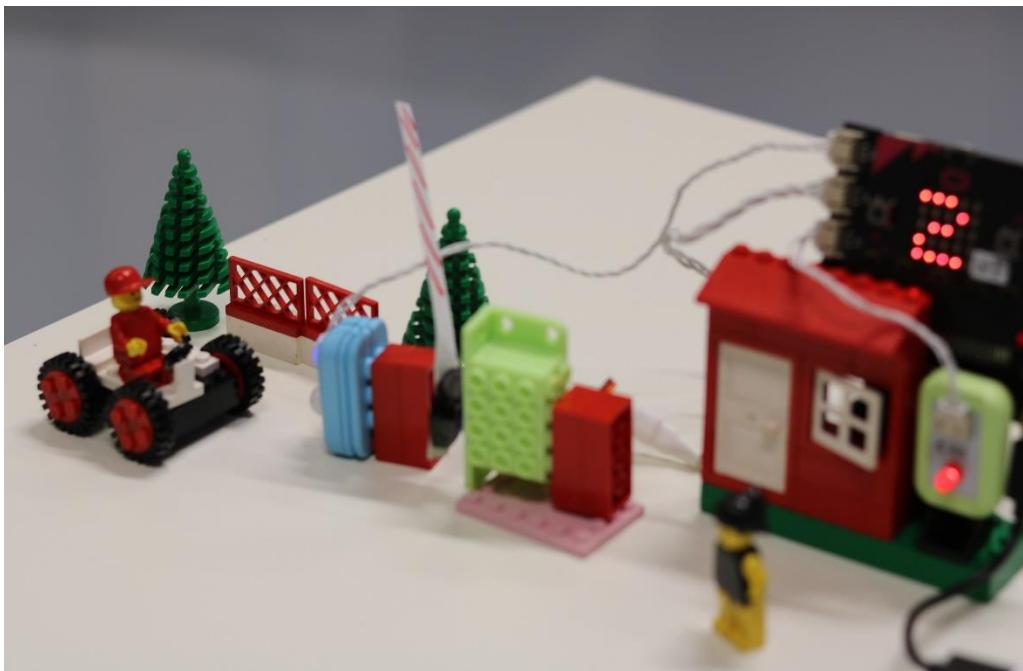


Kreativni kutak

Spuštena rampa:



Podignuta rampa 2. put:



Projekt 9: Roboalarm

Opis projekta

Izradite projekt Roboalarm i naučite kako u spoj s modulima iz Boson seta dodati i slušalice te kako micro:bitovi mogu međusobno komunicirati. Za izradu ovog projekta trebate 2 micro:bita koji će međusobno komunicirati, slušalice, svjetleću diodu i ventilator. Ideja je da prvi micro:bit bude robot čuvar pričvršćen za kutiju u kojoj se nalaze vaše važne i tajne stvari. Kad netko otvoriti kutijicu, prvi micro:bit javlja drugome da je pomaknut, a drugi micro:bit pokreće izlazne module koji su na njega spojeni: svjetleća dioda počne svijetliti, ventilator se kreće vrtjeti, a kroz slušalice se čuje zvuk alarma.

Prvi micro:bit zamaskirajte u robota. Izradite mu lice i tijelo od kartona, a inspirirajte se prijedlozima iz kreativnog kutka.

Potrebni moduli

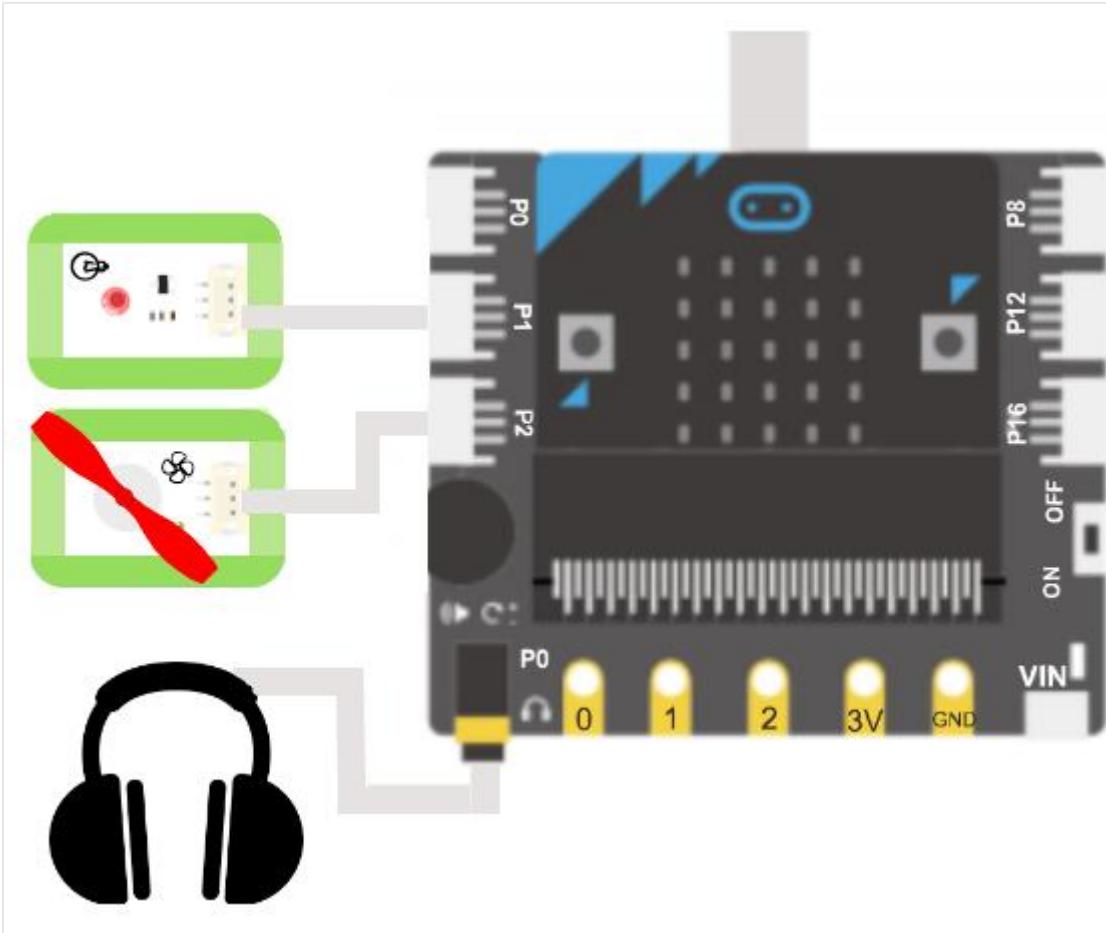
Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 2 kabela za spajanje modula, svjetleća dioda, ventilator, slušalice

Spajanje modula

Svjetleća dioda i ventilator su u ovom spoju digitalni izlazni moduli i možete ih spojiti na bilo koji izvod. U našem primjeru svjetleća dioda je spojena na izvod P1, a ventilator na izvod P2.

Slušalice spojite na predviđeno mjesto na pločici kao što je prikazano na slici spajanja. Volumen zvuka možete regulirati okretanjem potenciometra uz utor za slušalice.

Prikaz spajanja

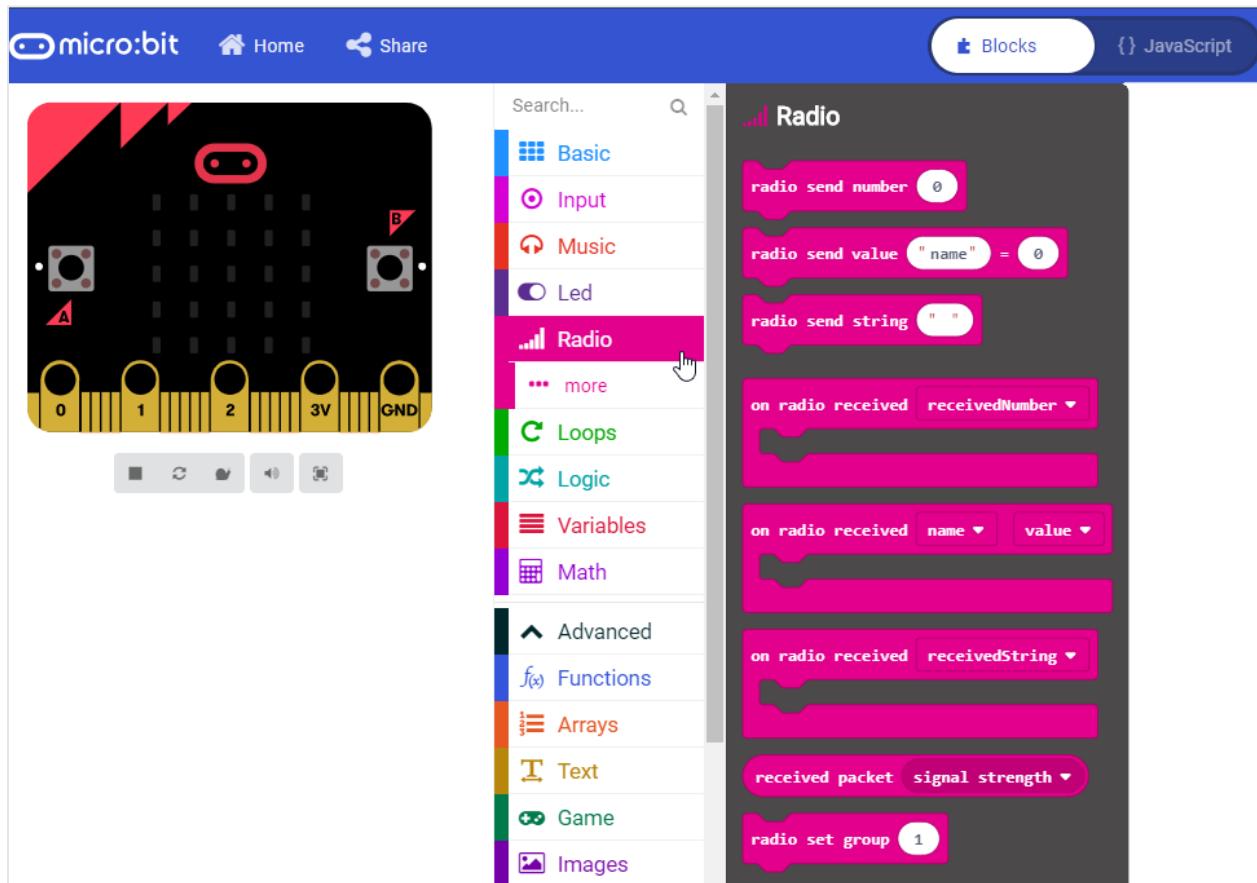


Izrada programa

Potrebne naredbe: *on start, radio set group, on screen up, on tilt left, radio send number, show icon, on radio received, if then else, operator usporedbe (jednako), digital write pin, start melody*

Najprije izradite program za prvi micro:bit (onaj koji čuva kutiju i šalje poruku da je pomaknut), a zatim izradite program za drugi micro:bit koji pokreće izlazne module spoja.

Da bi micro:bitovi mogli komunicirati potrebno je na početku programa postaviti kanal unutar kojeg oni izmjenjuju poruke. Sve naredbe za komunikaciju micro:bita pronaći ćete u kategoriji **Radio**.

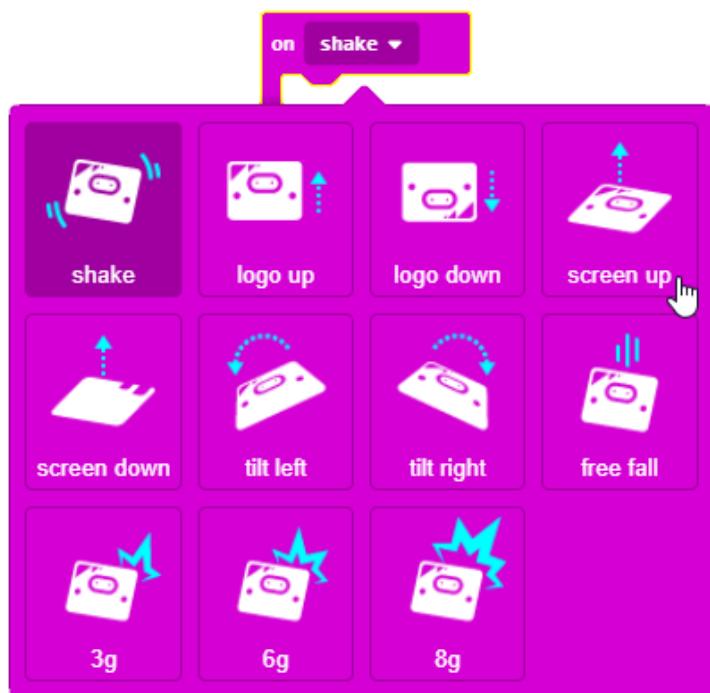


Iz kategorije **Radio** odaberite naredbu *radio set group 1* i smjestite ju u on start blok. Njime ste definirali da će micro:bitovi komunicirati kanalom 1.

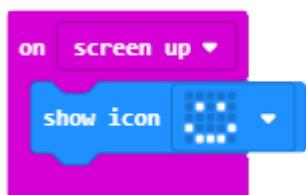


Iz kategorije **Input** odaberite *on shake* naredbu i smjestite ju u prostor na programiranje.

Pritisom na strelicu uz *shake* otvara se izbornik. Odaberite *screen up*.



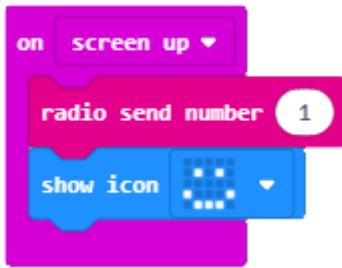
Unutar *on screen up* bloka smjeste ikonu nasmijanog lica (show icon iz kategorije **Basic**). Time ste definirali da se na zaslonu micro:bita prikazuje nasmijano lice kad je micro:bit u uspravnom položaju, a kutija zatvorena.



Kada je micro:bit uspravan neka drugome micro:bitu pošalje poruku u obliku broja. Za to upotrijebite naredbu *radio send number* iz kategorije **Radio**.

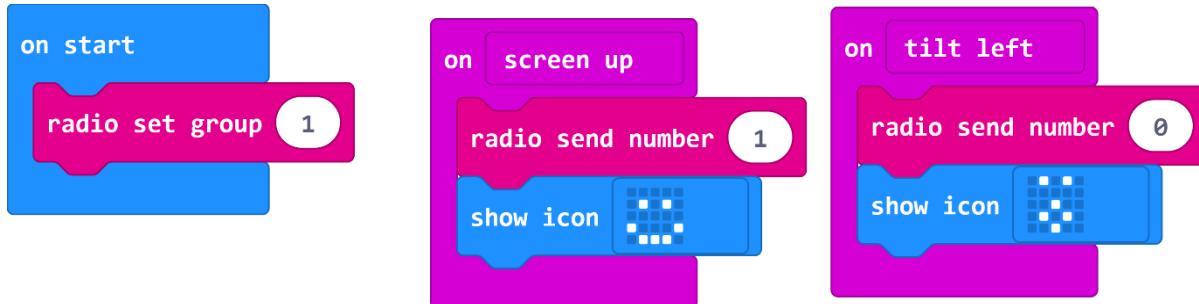


Tu naredbu smjestite u *on screen up* blok i upišite broj 1 umjesto 0.



Kopirajte ovaj cijeli blok i učinite u kopiranom bloku izmjene. Umjesto screen up, odaberite tilt left. Umjesto slanja broja 1 upišite broj 0, a umjesto nasmijanog lica postavite iznenađeno.

On tilt left blokom micro:bit će drugom micro:bitu poslati poruku koja sadrži vrijednost 0.



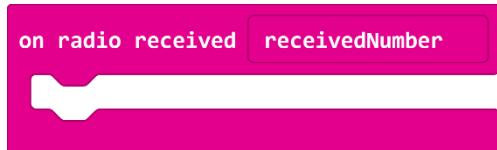
Time ste dovršili program za prvi micro:bit.

Slijedi izrada programa za drugi micro:bit.

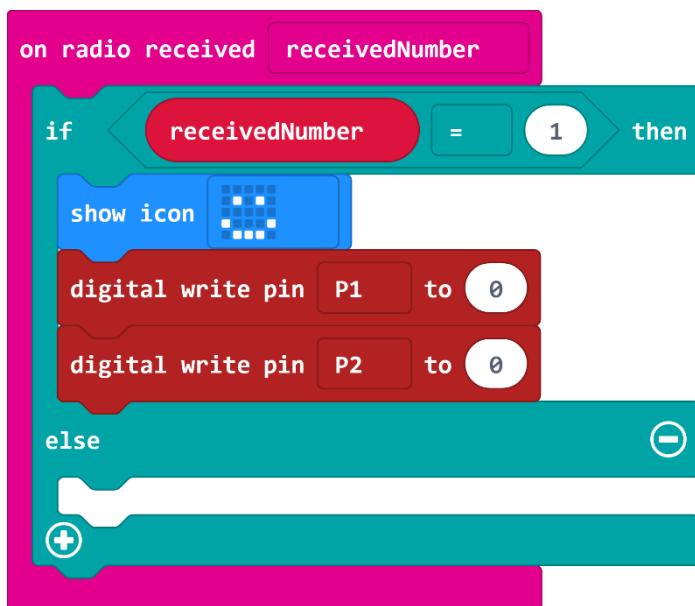
Drugi micro:bit također na početku programa mora imati definiran kanal komunikacije identičan onome za prvi micro:bit.



Sada slijedi dio programa kojim drugi micro:bit ispituje koja je vrijednost poruke koju prima. S obzirom na to da prima poruku u obliku broja iz kategorije Radio odaberite naredbu on radio received Receivednumber i postavite ju u prostor za programiranje.

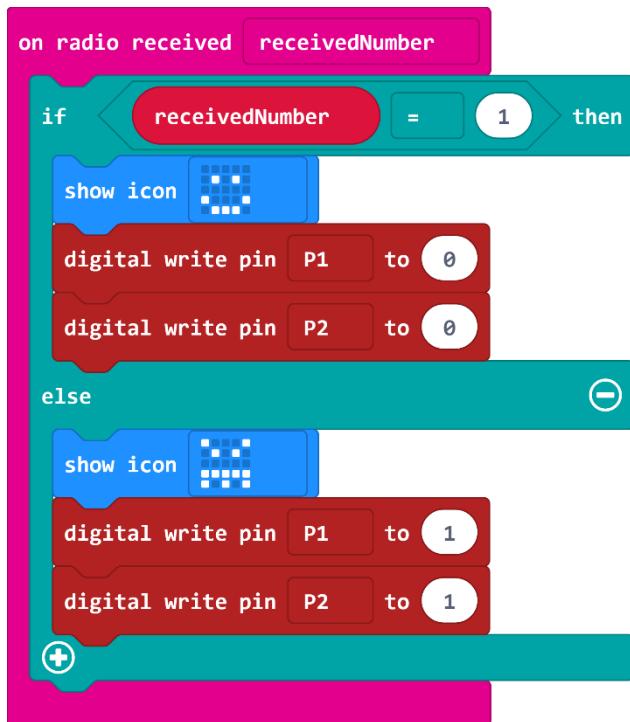


Unutar tog bloka postavite uvjet kojim ispitujete vrijednost primljene poruke. Ako je vrijednost primljene poruke jednaka 1, neka se na zaslonu micro:bita prikaže nasmijano lice (show icon happy), svjetleća dioda neka je ugašena (digital write pin P1 to 0) i ventilator ne radi (digital write pin P2 to 0).



Naredbu *receivedNumber* potražite u kategoriji **Variables**.

Inače, kad ovaj uvjet nije istinit, to znači da je kao poruku micro:bit primio neku drugu vrijednost (u ovom primjeru 1). Tada neka se na zaslonu micro:bita prikaže ljutito lice (show icon angry), svjetleća dioda neka se upali (digital write pin P1 to 1), a ventilator neka kreće s radom (digital write pin P2 to 0).



Još je potrebno dodati naredbe kojima će se reproducirati zvuk alarma preko slušalica. Njih možete pronaći u kategoriji **Music**.

The screenshot shows the micro:bit Scratch interface with the following details:

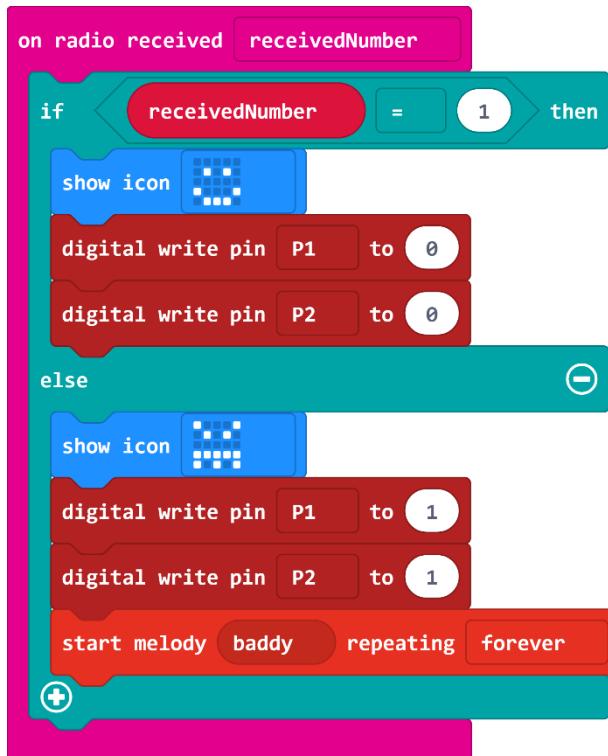
- Top Bar:** micro:bit, Home, Share, Blocks (selected), JavaScript.
- Scratch Stage:** Shows a micro:bit board with pins labeled 0, 1, 2, 3V, GND, and buttons A and B.
- Scratch Script Area:** An empty script area with a green hat block.
- Blocks Palette:**
 - Categories:** Basic, Input, **Music** (selected), Led, Radio, Loops, Logic, Variables, Math, Advanced, Functions, Arrays, Text, Game, Images, Pins, Serial.
 - Music Category Sub-blocks:**
 - play tone [Middle C] for [1 beat]
 - ring tone (Hz) [Middle C]
 - rest(ms) [1 beat]
 - start melody [dadadum] repeating [once]
 - music on [melody note played]
 - Middle C
 - [1 beat]
 - tempo (bpm)
 - change tempo by (bpm) [20]
 - set tempo to (bpm) [120]

Odaberite naredbu *start melody*.



U dijelu vrste melodije odaberite neku po želji, a u dijelu za ponavljanje (*repeating*) odaberite *forever*.

Smjestite naredbu u *else* dio na kraju. Time će alarm zvoniti dokle god je kutija otvorena.



Potrebno je u *then* dijelu definirati da alarm ne zvoni kad micro:bit primi poruku 1. U **Music** kategoriji nema naredbe za zaustavljanje sviranja melodije. Stoga ćete morati najprije odabratи *start melody* naredbu i smjestiti ju u *then* dio naredbe na kraju. S obzirom na to da je melodija zapravo niz tonova, u ovoj melodiji potrebno je u parametru vrste melodije postaviti prazan niz. Kako biste to učinili u editoru se prebacite u **JavaScript** način rada.

Prikazat će se programski kôd u **JavaSriptu**. U 6. redu koda, u naredbi music.beginMelody obrišite *music.builtMelody(Melodies.Dadadum)* i upišite dvije uglate zagrade.

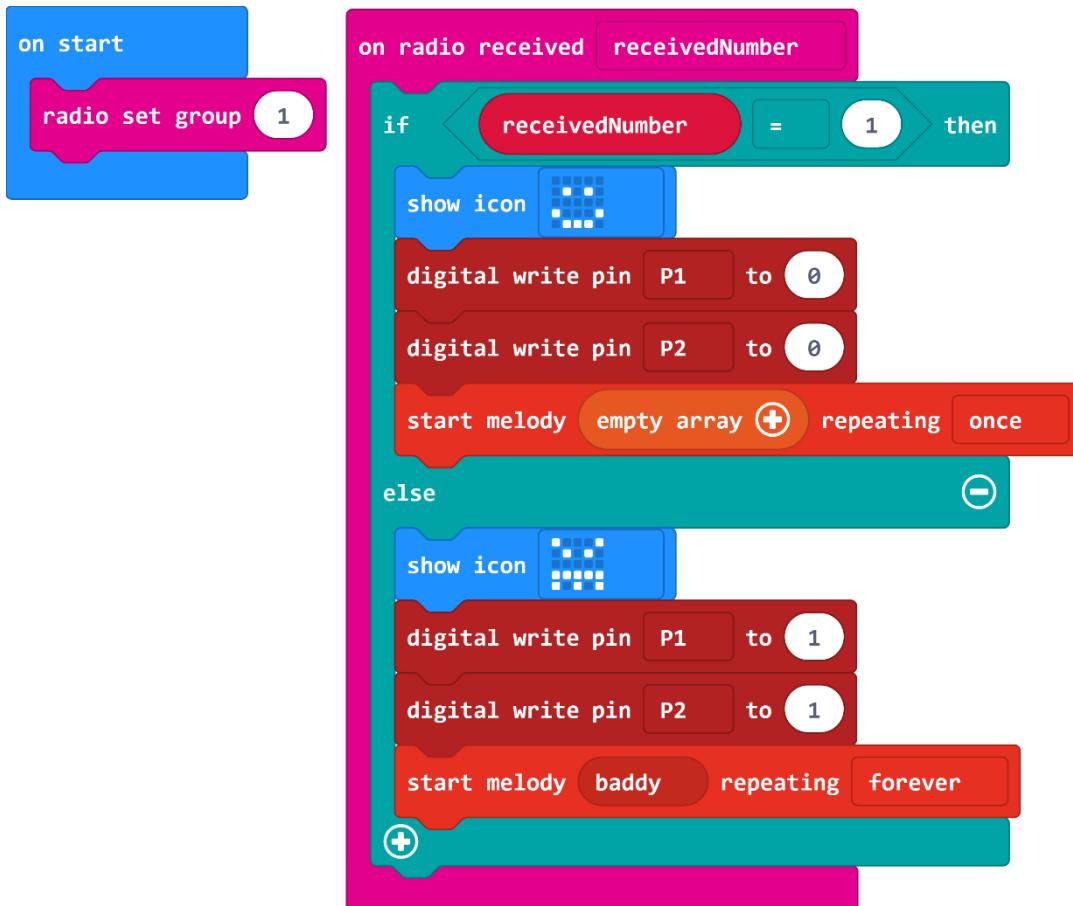
Blocks {} JavaScript

```

1 radio.onReceivedNumber(function (receivedNumber) {
2     if (receivedNumber == 1) {
3         basic.showIcon(IconNames.Happy)
4         pins.digitalWritePin(DigitalPin.P1, 0)
5         pins.digitalWritePin(DigitalPin.P2, 0)
6         music.beginMelody([], MelodyOptions.Once)
7     } else {
8         basic.showIcon(IconNames.Angry)
9         pins.digitalWritePin(DigitalPin.P1, 1)
10        pins.digitalWritePin(DigitalPin.P2, 1)
11        music.beginMelody(music.builtInMelody(Melodies.Baddy), MelodyOptions.Forever)
12    }
13 })
14

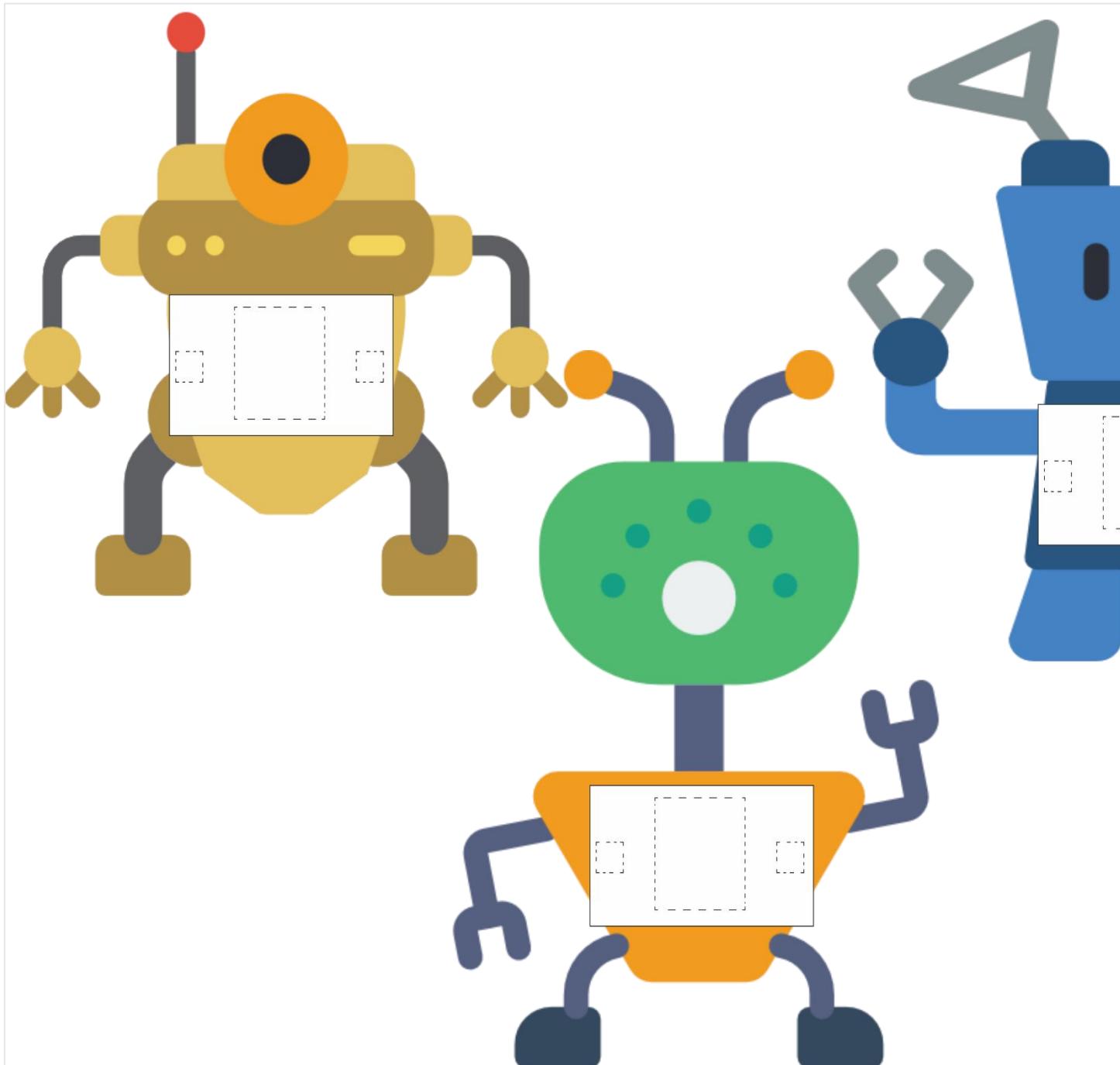
```

Vratite se u **Blocks** način rada i vidjet ćete da je u *naredbi start melody* prazan niz. Time ste dovršili program za ovaj projekt.

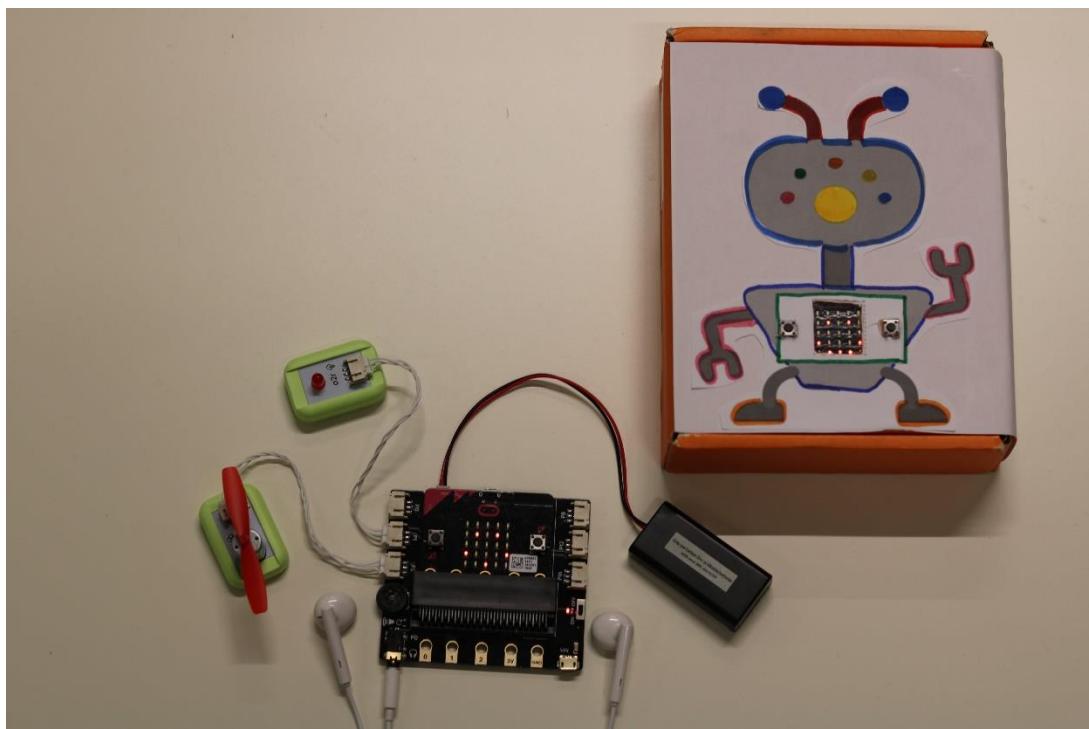


Kreativni kutak

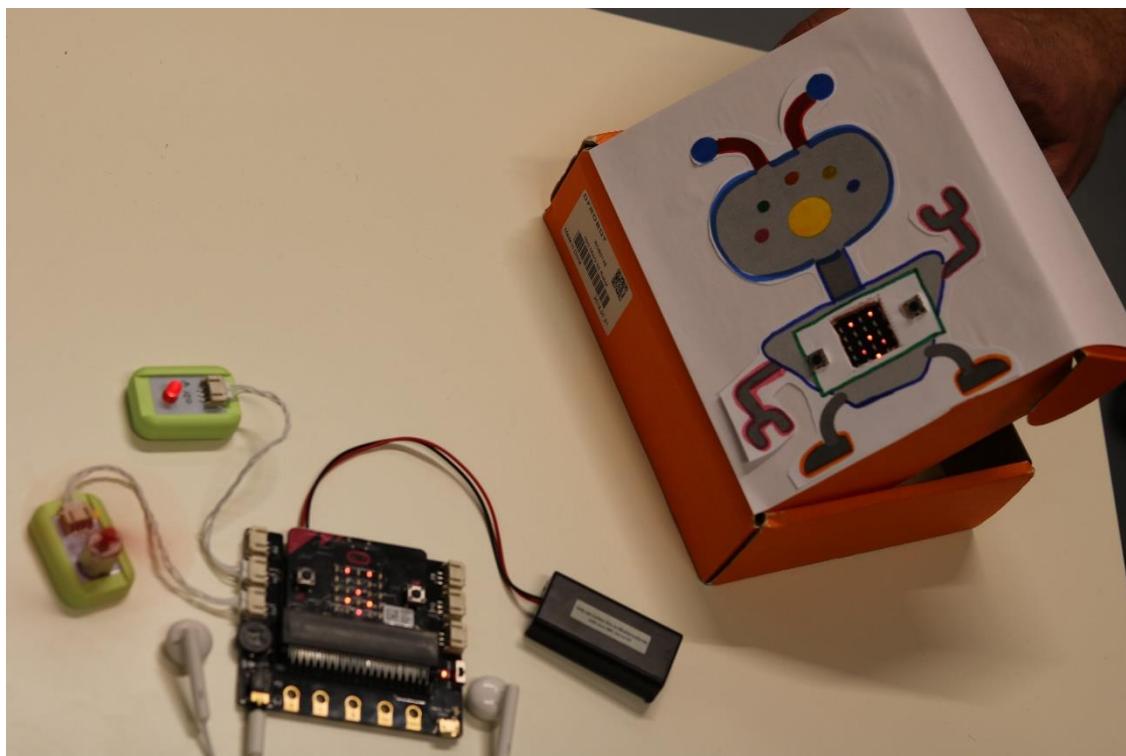
Odaberite model za svog robota:



Zatvorena kutija:



Otvorena kutija:



Projekt 10: Robot – kreiraj ga micro:bit kockicom

Opis projekta

U ovom projektu imate priliku primijeniti svoje znanje o micro:bitu i servo motoru te pomoći njih izraditi vlastitog robota koji se kreće. Za ovaj projekt bi bilo idealno da imate 2 micro:bita, ali možete raditi i samo s jednim tako da izmjenjujete programe na njemu i postepeno rješavate zadatak. Prvi micro:bit najprije pretvorite u igraču kockicu. To znači da svaki put kad ga protresete na njemu će se prikazati slučajno odabrani broj od 1 do 6. Protresanjem micro:bita kockice tako možete odrediti izgled svog robota. U kreativnom kutku ovog projekta možete pronaći 6 različitih mogućnosti izgleda robota (glava, tijelo i ruke). Prvim protresanjem odredi glavu, drugim tijelo, a trećim robotske ruke. Robota izradite od kartona, kolaža i ljepila te upotrijebite komad žice ili špage za povlačenje stražnjeg dijela robota prema naprijed kako bi se kretao.

Za kretanje robota potreban je servo motor iz Boson seta. Njegovim pomicanjem upravlјat će drugi micro:bit koji ćete smjestiti u pločicu za spajanje. Pritisom na tipku A neka robot krene s kretanjem, a pritiskom na tipku B neka se zaustavi.

Prvi micro:bit je odradio dio posla za određivanje izgleda robota, a sada ga možete upotrijebiti za njegovo lice tako da se na njemu prikazuje robotovo raspoloženje. Kakvog raspoloženja će biti vaš robot odredite sami.

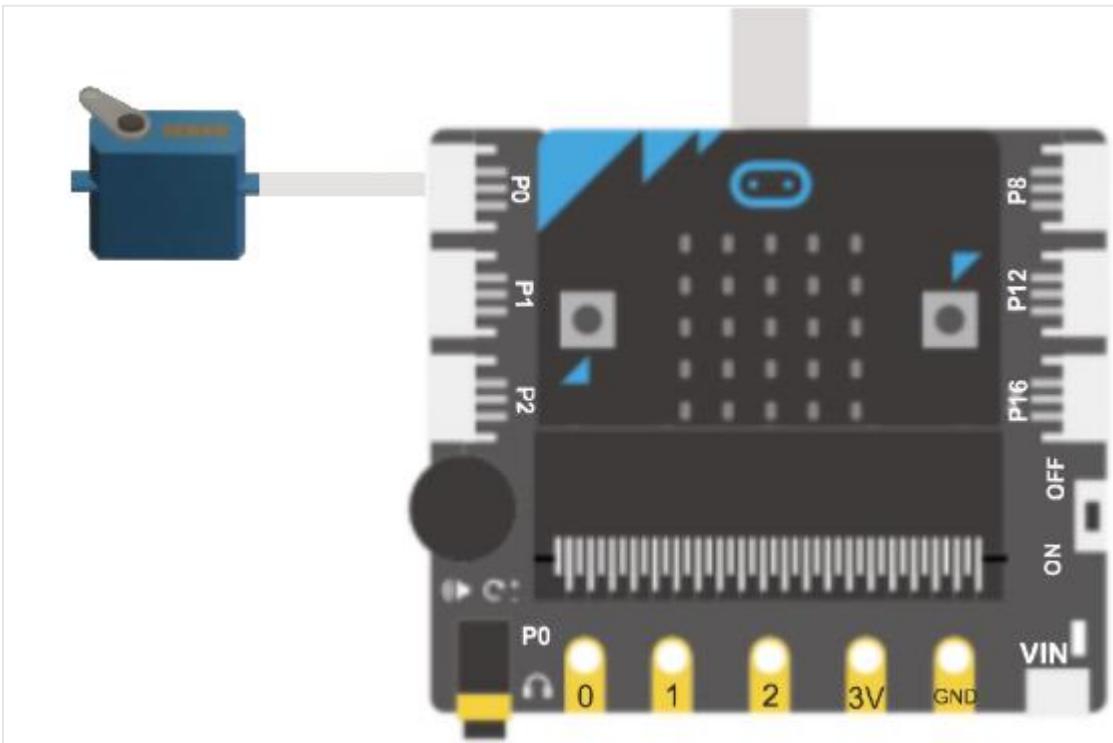
Potrebni moduli

Pločica za spajanje elektroničkih modula, USB kabel za napajanje, 1 kabela za spajanje modula, servo motor

Spajanje modula

U pločicu smjestite micro:bit. U ovom spajanju upotrijebite samo servo motor koji ćete spojiti na izvod P0.

Prikaz spajanja



Izrada programa

Micro:bit kockica

Potrebne naredbe: *on shake, show number, pick random*

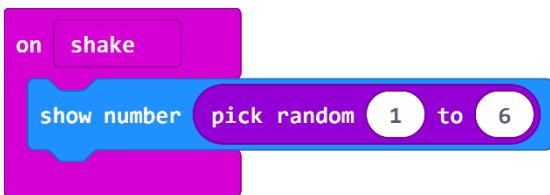
Prvi micro:bit će biti kockica čijim protresanjem određujete kako će robot izgledati. Za protresanje micro:bita upotrijebite naredbu *on shake* iz kategorije **Input**.



Kad se micro:bit protrese mora se na zaslonu prikazati broj. Za prikaz broja odaberite naredbu *show number* iz kategorije **Basic** smjestite ju u *on shake* blok.



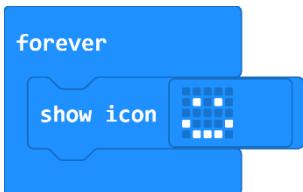
Trenutnim programskim kôdom micro:bit prikazuje samo nulu kad ga se protrese. Prema zadatku on mora prikazati slučajno odabrane brojeve od 1 do 6. Da bi se to dogodilo kao argument naredbe show number postavite naredbu *pick random 0 to 10*. Promijenite argumente 0 i 10 u 1 i 6 kako bi se protresanjem micro:bita prikazali brojevi u tom rasponu.



Micro:bit robot – lice

Potrebne naredbe: *forever, show icon*

Nakon što je prvi micro:bit odradio svoj posao kockice, sada na njega možete prebaciti program kojim će se na zaslonu micro:bita prikazivati raspoloženje robota. Raspoloženje možete nacrtati sami koristeći *show leds* naredbu ili koristiti neku gotovu ikonu iz naredbe *show icon*. Obje naredbe potražite u kategoriji **Basic**.



Micro:bit robot – kretanje

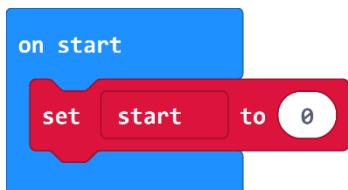
Potrebne naredbe: *on start, set variable to, on button A pressed, on button B pressed, forever, if then else, operator usporedbe, pause (ms), servo write pin P0 to*

Odredili ste izgled robota i njegov izraz lica, a sada slijedi izrada programa za drugi micro:bit koji će upravljati kretanjem robota.

Najprije iz kategorije Basic odaberite **on start** naredbu i smjestite ju u prostor za programiranje.

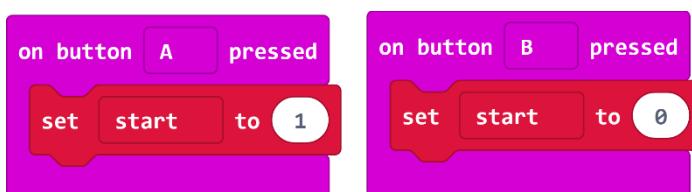


Na početku programa, robot mora stajati prije nego pritisnete tipku A. Jednom kad se robot počne kretati zaustavlja pritiskom na tipku B te u bio kojem trenutku može ponovno krenuti ako se pritisne tipka A. Da bi to moglo funkcionirati, potrebno je da u kategoriji **Variables** kreirate varijablu koju možete nazvati *start*. U *on start* bloku inicijalizirajte varijablu *start* na 0.

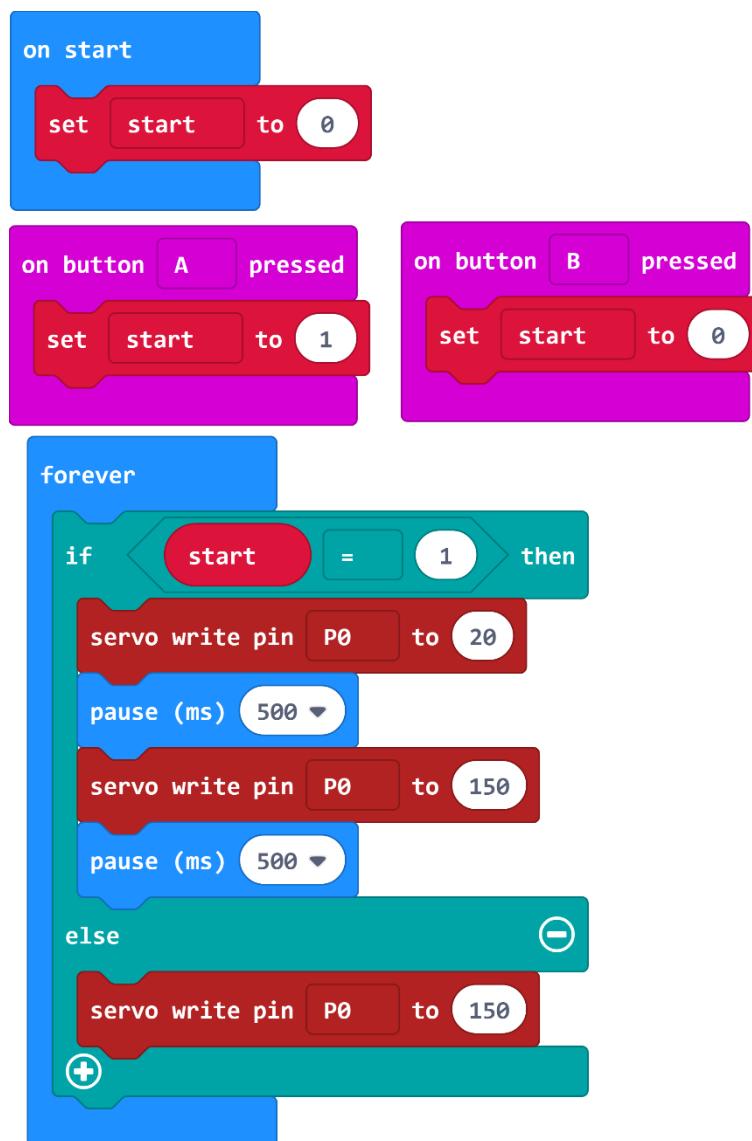


Provjerom stanja varijable *start* u *forever* petlji, robot će se kretati ili mirovati. Ako je njezina vrijednost 0 (kao na početku programa), robot stoji. Ako joj je vrijednost različita od 0 (primjerice 1), robot se kreće.

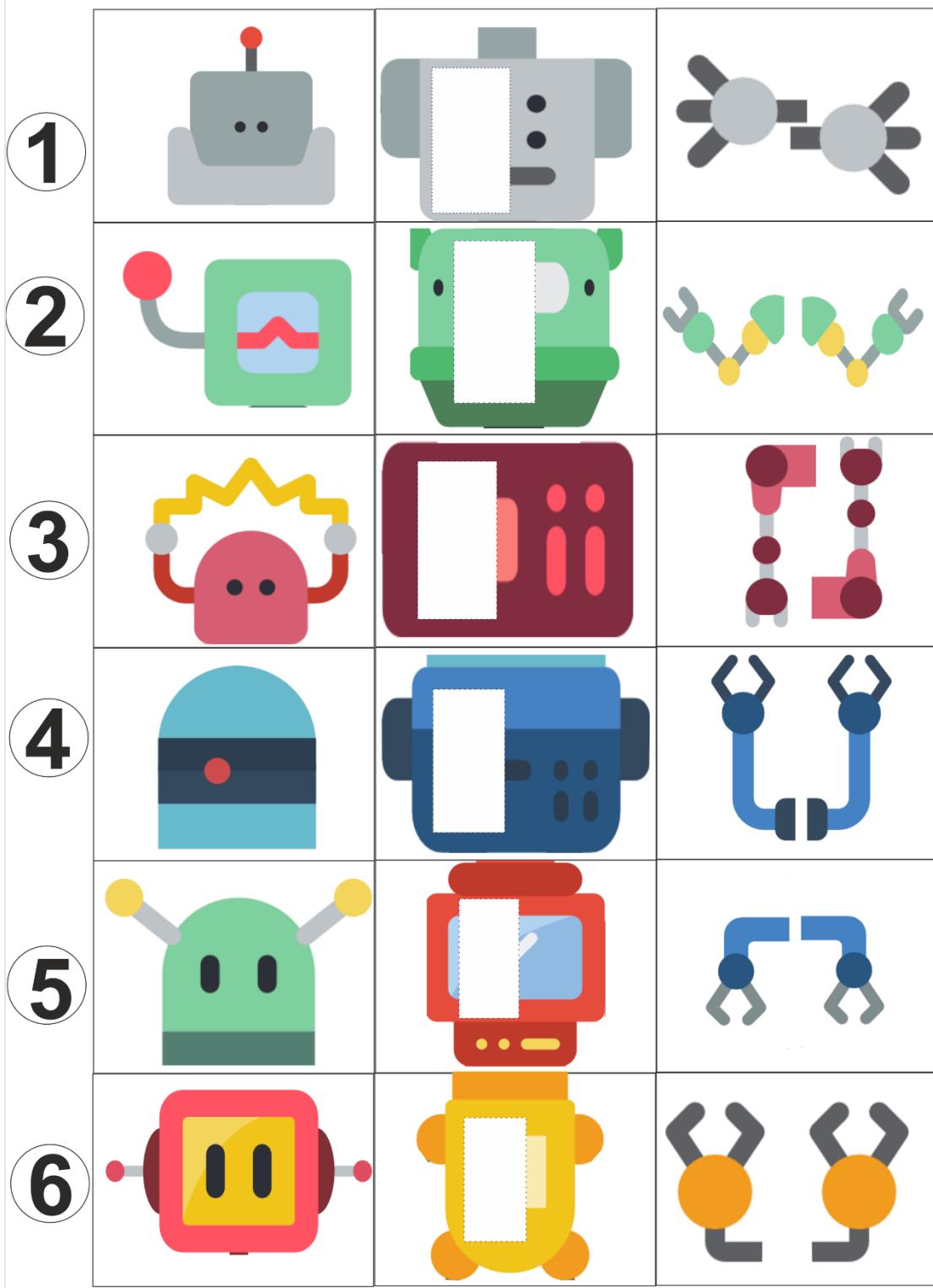
Vrijednost varijable *start* mijenjajte pritiskom na tipke A i B. Kad se pritisne tipka A, varijabla *start* neka poprimi vrijednost 1, a kad se pritisne tipka B, varijabla *start* neka ponovno postane 0.



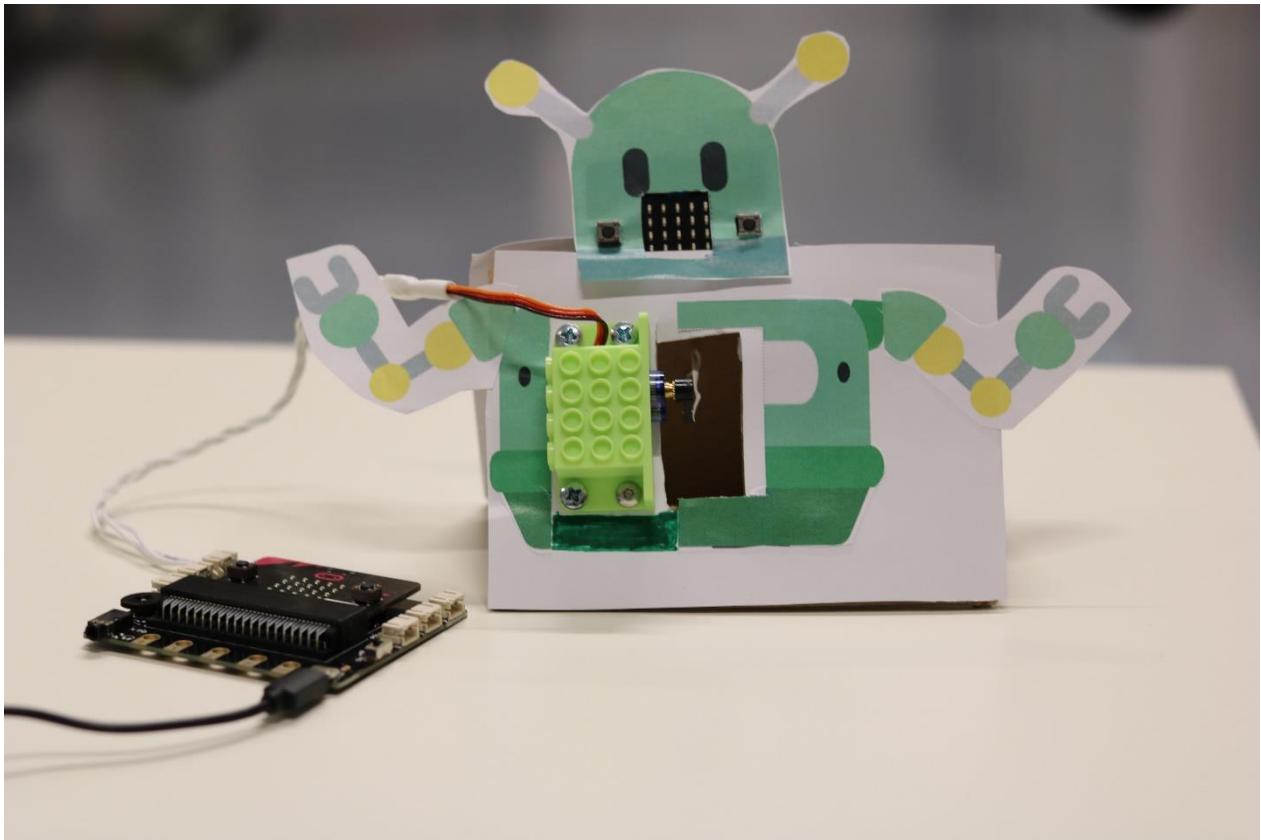
Sada u *forever* petlji provjeravajte stanje varijable *start*. Ako je vrijednost varijable *start* jednaka 1., servo motor neka krene s radom pomicanjem polugice naprijed-nazad. Inače, ako vrijednost varijable *start* nije 1, servo motor neka se zaustavi u određenom položaju. Za postavljanje uvjeta koristite naredbu *if then else* iz kategorije **Logic**. Za rad servo motora upotrijebite naredbu *servo write pin P0 to* iz kategorije **Pins**. Želite da servo motor okreće polugicu naprijed-nazad, stoga u *then* dijelu programa postavite dvije pozicije servo motora i između njih naredbu za pauzu kako bi servo motoru dali dovoljno vremena da dođe iz jedne pozicije na drugu. U *else* dijelu odredite naredbom *servo write pin P0 to* u kojoj poziciji da bude polugica motora kad se robot zaustavi.



Kreiraj svog robota



Ovakvog robota smo mi izradili:



Presavinite karton na dva jednaka dijela.

U sredini kartona izrežite rupu veličine 5x2.5 cm kako bi se polugica mogla nesmetano pomicati.

Na kartoni iscrtajte ili kolaž papirom uredite izgled svog robot te zaliđepite ljepilom na karton.

Vijcima i maticama (veličina M4) pričvrstite servo motor za karton (ili ga zaliđepite selotejpom).

Na polugicu pričvrstite jedan kraj žice ili špage, a drugi kraj za stražnji dio kartona kako bi ga polugica mogla pomicati prema naprijed.